

501P1118US60

4

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application: 2000年 8月17日

出 願 番 号

Application Number: 特願2000-247461

出 願 人

Applicant(s): ソニー株式会社

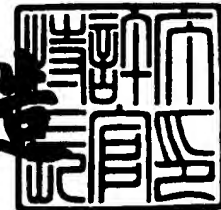


CERTIFIED COPY OF  
PRIORITY DOCUMENT

2001年 5月11日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3038664

【書類名】 特許願

【整理番号】 00006933

【提出日】 平成12年 8月17日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 19/00

【発明の名称】 データ処理装置およびデータ処理方法、並びにプログラム提供媒体

【請求項の数】 18

【発明者】

    【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社  
    内

    【氏名】 岡上 拓己

【特許出願人】

    【識別番号】 000002185

    【氏名又は名称】 ソニー株式会社

    【代表者】 出井 伸之

【代理人】

    【識別番号】 100101801

    【弁理士】

    【氏名又は名称】 山田 英治

    【電話番号】 03-5541-7577

【選任した代理人】

    【識別番号】 100093241

    【弁理士】

    【氏名又は名称】 宮田 正昭

    【電話番号】 03-5541-7577

【選任した代理人】

    【識別番号】 100086531

    【弁理士】

【氏名又は名称】 澤田 俊夫

【電話番号】 03-5541-7577

【先の出願に基づく優先権主張】

【出願番号】 特願2000-222125

【出願日】 平成12年 7月24日

【手数料の表示】

【予納台帳番号】 062721

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9904833

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ処理装置およびデータ処理方法、並びにプログラム提供媒体

【特許請求の範囲】

【請求項 1】

記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行するデータ処理装置であり、

前記検証値は、コンテンツのカテゴリ毎に独立した検証値として生成して格納する構成を有することを特徴とするデータ処理装置。

【請求項 2】

前記データ処理装置は、

コンテンツ利用に際して、利用対象コンテンツ構成データに基づいて検証値を算出し、該算出検証値と、予め格納された検証値との比較処理を実行し、一致する場合にのみコンテンツ利用を可能とした構成を有することを特徴とする請求項 1 に記載のデータ処理装置。

【請求項 3】

前記記憶装置は、複数のディレクトリの各々に異なるカテゴリのコンテンツデータを格納する構成を有し、

前記検証値は、前記複数のディレクトリの各々を単位とするコンテンツの集合に対して生成される値であることを特徴とする請求項 1 に記載のデータ処理装置。

【請求項 4】

前記記憶装置は、フラッシュメモリであり、

前記カテゴリ毎の検証値は、フラッシュメモリの使用禁止ブロックとして設定された領域に格納される構成を有することを特徴とする請求項 1 に記載のデータ処理装置。

【請求項 5】

前記カテゴリは、コンテンツの種類に基づいて設定された構成であり、

コンテンツの種類別に独立した検証値を設定し、格納する構成を有することを特徴とする請求項 1 に記載のデータ処理装置。

【請求項 6】

前記カテゴリは、コンテンツの暗号処理鍵として設定されるコンテンツキー K c o n を暗号化して提供する有効化キープブロック (E K B) の管理エンティティに基づいて設定された構成であり、

有効化キープブロック (E K B) の管理エンティティ別に独立した検証値を設定し、格納する構成を有することを特徴とする請求項 1 に記載のデータ処理装置。

【請求項 7】

前記検証値は、検証対象となるコンテンツおよびヘッダ情報等のコンテンツ関連データを構成する部分データをメッセージとして D E S 暗号処理によって生成されるメッセージ認証符号 (M A C) に基づいて生成されるデータであることを特徴とする請求項 1 に記載のデータ処理装置。

【請求項 8】

記憶装置に格納されるコンテンツまたはヘッダ情報中の改竄有無検証データとしてのメッセージ認証符号 (M A C) を生成して格納するデータ処理装置であり、

コンテンツまたはヘッダ情報中の異なるデータ領域を M A C 生成対象として複数のメッセージ認証符号 (M A C) を生成するとともに、前記複数のメッセージ認証符号 (M A C) の各 M A C 生成対象データ領域の一部を共通データとし、いずれかの M A C 更新時に、前記共通データを更新し、他方の M A C 更新処理も併せて実行する構成としたことを特徴とするデータ処理装置。

【請求項 9】

記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行するデータ処理方法であり、

前記検証値を、コンテンツのカテゴリ毎に独立した検証値として生成して格納することを特徴とするデータ処理方法。

【請求項 1 0】

前記データ処理方法において、さらに、

コンテンツ利用に際して、利用対象コンテンツ構成データに基づいて検証値を算出し、該算出検証値と、予め格納された検証値との比較処理を実行し、一致する場合にのみコンテンツ利用を行なうことを特徴とする請求項 9 に記載のデータ処理方法。

【請求項 1 1】

前記記憶装置は、複数のディレクトリの各々に異なるカテゴリのコンテンツデータを格納する構成を有し、

前記検証値は、前記複数のディレクトリの各々を単位とするコンテンツの集合に対して生成することを特徴とする請求項 9 に記載のデータ処理方法。

【請求項 1 2】

前記記憶装置は、フラッシュメモリであり、

前記カテゴリ毎の検証値は、フラッシュメモリの使用禁止ブロックとして設定された領域に格納することを特徴とする請求項 9 に記載のデータ処理方法。

【請求項 1 3】

前記カテゴリは、コンテンツの種類に基づいて設定され、

コンテンツの種類別に独立した検証値を設定し、格納することを特徴とする請求項 9 に記載のデータ処理方法。

【請求項 1 4】

前記カテゴリは、コンテンツの暗号処理鍵として設定されるコンテンツキー K c o n を暗号化して提供する有効化キーブロック (E K B) の管理エンティティに基づいて設定され、

有効化キーブロック (E K B) の管理エンティティ別に独立した検証値を設定し、格納することを特徴とする請求項 9 に記載のデータ処理方法。

【請求項 1 5】

前記検証値は、検証対象となるコンテンツおよびヘッダ情報等のコンテンツ関連データを構成する部分データをメッセージとして D E S 暗号処理によって生成されるメッセージ認証符号 (M A C) に基づいて生成されるデータであることを特徴とする請求項 9 に記載のデータ処理方法。

## 【請求項 1 6】

前記検証対象となるコンテンツおよびヘッダ情報には、異なるデータ領域をMAC生成対象とした複数のメッセージ認証符号(MAC)を含み、

前記複数のメッセージ認証符号(MAC)は、各々のMAC生成対象データ領域の一部を共通とし、いずれかのMAC更新時に他方のMAC更新処理も併せて実行することを特徴とする請求項 1 5 に記載のデータ処理方法。

## 【請求項 1 7】

記憶装置に格納されるコンテンツまたはヘッダ情報中の改竄有無検証データとしてのメッセージ認証符号(MAC)を生成して格納するデータ処理方法であり、

コンテンツまたはヘッダ情報中の異なるデータ領域をMAC生成対象として複数のメッセージ認証符号(MAC)を生成するとともに、前記複数のメッセージ認証符号(MAC)の各MAC生成対象データ領域の一部を共通データとし、いずれかのMAC更新時に、前記共通データを更新し、他方のMAC更新処理も併せて実行することを特徴とするデータ処理方法。

## 【請求項 1 8】

記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行するデータ処理をコンピュータ・システム上で実行せしめるコンピュータ・プログラムを提供するプログラム提供媒体であって、前記コンピュータ・プログラムは、

検証値を、コンテンツのカテゴリ毎に独立した検証値として生成して格納するステップを有することを特徴とするプログラム提供媒体。

## 【発明の詳細な説明】

## 【0 0 0 1】

## 【発明の属する技術分野】

本発明は、データ処理装置およびデータ処理方法、並びにプログラム提供媒体に関する。特に、記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行する構成において、検証値をコンテンツのカテゴリ毎に独立した検証値として生成して格

納する構成とし、コンテンツ改竄チェック処理の効率化を実現したデータ処理装置およびデータ処理方法、並びにプログラム提供媒体に関する。

#### 【0002】

##### 【従来の技術】

昨今、音楽データ、ゲームプログラム、画像データ等、様々なソフトウェアデータ（以下、これらをコンテンツ（Content）と呼ぶ）を、インターネット等のネットワーク、あるいは、メモリカード、DVD、CD等の流通可能な記憶媒体を介して流通させるコンテンツ流通が盛んになってきている。これらの流通コンテンツは、ユーザの所有するPC（Personal Computer）、再生専用器、あるいはゲーム機器におけるコンテンツデータの受信、あるいはメモリカード、CD、DVD等の記憶媒体の装着により、コンテンツ再生処理が実行されたり、あるいは外部からの入力コンテンツを再生器、PC等に内蔵の記録デバイス、例えばメモリカード、ハードディスク等に格納し、再度、格納媒体から再生する等の方法により利用される。

#### 【0003】

再生装置、ゲーム機器、PC等の情報機器には、流通コンテンツをネットワークから受信するため、あるいはDVD、CD等にアクセスするためのインタフェースを有し、さらにコンテンツの再生に必要なとなる制御手段、プログラム、データのメモリ領域として使用されるRAM、ROM等を有する。

#### 【0004】

音楽データ、画像データ、あるいはプログラム等の様々なコンテンツは、再生機器として利用される再生装置、ゲーム機器、PC等の情報機器本体からのユーザ指示、あるいは接続された入力手段を介したユーザの指示により、例えば内蔵、あるいは着脱自在の記憶媒体から呼び出され、情報機器本体、あるいは接続されたディスプレイ、スピーカ等を通じて再生される。

#### 【0005】

ゲームプログラム、音楽データ、画像データ等、多くのソフトウェア・コンテンツは、一般的にその作成者、販売者に頒布権等が保有されている。従って、これらのコンテンツの配布に際しては、一定の利用制限、すなわち、正規なユーザ

に対してのみ、ソフトウェアの使用を許諾し、許可のない複製等が行われないようにする、すなわちセキュリティを考慮した構成をとるのが一般的となっている。

#### 【0006】

ユーザに対する利用制限を実現する1つの手法が、配布コンテンツの暗号化処理である。すなわち、例えばインターネット等を介して暗号化された音声データ、画像データ、ゲームプログラム等の各種コンテンツを配布するとともに、正規ユーザであると確認された者に対してのみ、配布された暗号化コンテンツを復号する手段、すなわち復号鍵を付与する構成である。

#### 【0007】

暗号化データは、所定の手続きによる復号化処理によって利用可能な復号データ（平文）に戻すことができる。このような情報の暗号化処理に暗号化鍵を用い、復号化処理に復号化鍵を用いるデータ暗号化、復号化方法は従来からよく知られている。

#### 【0008】

暗号化鍵と復号化鍵を用いるデータ暗号化・復号化方法の態様には様々な種類があるが、その1つの例としていわゆる共通鍵暗号化方式と呼ばれている方式がある。共通鍵暗号化方式は、データの暗号化処理に用いる暗号化鍵とデータの復号化に用いる復号化鍵を共通のものとして、正規のユーザにこれら暗号化処理、復号化に用いる共通鍵を付与して、鍵を持たない不正ユーザによるデータアクセスを排除するものである。この方式の代表的な方式にDES（データ暗号標準：Data encryption standard）がある。

#### 【0009】

上述の暗号化処理、復号化に用いられる暗号化鍵、復号化鍵は、例えばあるパスワード等に基づいてハッシュ関数等の一方向性関数を適用して得ることができる。一方向性関数とは、その出力から逆に入力を求めるのは非常に困難となる関数である。例えばユーザが決めたパスワードを入力として一方向性関数を適用して、その出力に基づいて暗号化鍵、復号化鍵を生成するものである。このようにして得られた暗号化鍵、復号化鍵から、逆にそのオリジナルのデータであるパス

ワードを求めることは実質上不可能となる。

【 0 0 1 0 】

また、暗号化するときに使用する暗号化鍵による処理と、復号するときに使用する復号化鍵の処理とを異なるアルゴリズムとした方式がいわゆる公開鍵暗号化方式と呼ばれる方式である。公開鍵暗号化方式は、不特定のユーザが使用可能な公開鍵を使用する方法であり、特定個人に対する暗号化文書を、その特定個人が発行した公開鍵を用いて暗号化処理を行なう。公開鍵によって暗号化された文書は、その暗号化処理に使用された公開鍵に対応する秘密鍵によってのみ復号処理が可能となる。秘密鍵は、公開鍵を発行した個人のみが所有するので、その公開鍵によって暗号化された文書は秘密鍵を持つ個人のみが復号することができる。公開鍵暗号化方式の代表的なものには R S A (Rivest-Shamir-Adleman) 暗号がある。このような暗号化方式を利用することにより、暗号化コンテンツを正規ユーザに対してのみ復号可能とするシステムが可能となる。

【 0 0 1 1 】

【発明が解決しようとする課題】

コンテンツデータの正当性、すなわちデータが改竄されていないことをチェックするために検証用のチェック値を正規のコンテンツデータに基づいて生成して予めメモリに格納し、コンテンツ利用時に検証対象のデータに基づいて生成したチェック値と格納チェック値とを照合処理することによって、データ検証を行なう方法が従来から行なわれている。

【 0 0 1 2 】

しかしながら、メモリに格納するコンテンツ数が増大すると、検証用のチェック値を正規のコンテンツデータに基づいて生成し、格納し管理することが困難となる。特に、昨今フラッシュメモリを使用したメモリカード等の容量の大きい媒体においては、音楽データ、画像データ、プログラムデータ等、様々なカテゴリのコンテンツデータがメモリに格納されることとなる。このような環境においては、コンテンツのチェック値の生成処理、格納処理、改竄チェック処理の管理は困難となる。格納データ全体に対するチェック値を生成すると、チェック対象となったデータ全体に対するチェック値生成処理を実行することが必要となる。例

例えばDES-CBCモードにおいて生成されるメッセージ認証符号（MAC）により、チェック値ICVを求める手法を行なう場合、データ全体に対するDES-CBCの処理を実行することが必要となる。この計算量は、データ長が長くなるにつれ増大することとなり、処理効率の点で問題がある。

## 【0013】

本発明は、このような従来技術の問題点を解決するものであり、データ正当性の確認処理を効率的に実行し、コンテンツデータの検証処理の効率化、さらに検証後の記録デバイスに対するダウンロード処理、あるいは検証後の再生処理等を効率的に実行することを可能とするデータ処理装置およびデータ処理方法、並びにプログラム提供媒体を提供することを目的とする。

## 【0014】

## 【課題を解決するための手段】

本発明の第1の側面は、

記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行するデータ処理装置であり、

前記検証値は、コンテンツのカテゴリ毎に独立した検証値として生成して格納する構成を有することを特徴とするデータ処理装置にある。

## 【0015】

さらに、本発明のデータ処理装置の一実施態様において、前記データ処理装置は、コンテンツ利用に際して、利用対象コンテンツ構成データに基づいて検証値を算出し、該算出検証値と、予め格納された検証値との比較処理を実行し、一致する場合にのみコンテンツ利用を可能とした構成を有することを特徴とする。

## 【0016】

さらに、本発明のデータ処理装置の一実施態様において、前記記憶装置は、複数のディレクトリの各々に異なるカテゴリのコンテンツデータを格納する構成を有し、前記検証値は、前記複数のディレクトリの各々を単位とするコンテンツの集合に対して生成される値であることを特徴とする。

## 【0017】

さらに、本発明のデータ処理装置の一実施態様において、前記記憶装置は、フラッシュメモリであり、前記カテゴリ毎の検証値は、フラッシュメモリの使用禁止ブロックとして設定された領域に格納される構成を有することを特徴とする。

## 【 0 0 1 8 】

さらに、本発明のデータ処理装置の一実施態様において、前記カテゴリは、コンテンツの種類に基づいて設定された構成であり、コンテンツの種類別に独立した検証値を設定し、格納する構成を有することを特徴とする。

## 【 0 0 1 9 】

さらに、本発明のデータ処理装置の一実施態様において、前記カテゴリは、コンテンツの暗号処理鍵として設定されるコンテンツキー *K c o n* を暗号化して提供する有効化キーブロック ( *E K B* ) の管理エンティティに基づいて設定された構成であり、有効化キーブロック ( *E K B* ) の管理エンティティ別に独立した検証値を設定し、格納する構成を有することを特徴とする。

## 【 0 0 2 0 】

さらに、本発明のデータ処理装置の一実施態様において、前記検証値は、検証対象となるコンテンツおよびヘッダ情報等のコンテンツ関連データを構成する部分データをメッセージとして *D E S* 暗号処理によって生成されるメッセージ認証符号 ( *MAC* ) に基づいて生成されるデータであることを特徴とする。

## 【 0 0 2 1 】

さらに、本発明の第 2 の側面は、

記憶装置に格納されるコンテンツまたはヘッダ情報中の改竄有無検証データとしてのメッセージ認証符号 ( *MAC* ) を生成して格納するデータ処理装置であり

コンテンツまたはヘッダ情報中の異なるデータ領域を *MAC* 生成対象として複数のメッセージ認証符号 ( *MAC* ) を生成するとともに、前記複数のメッセージ認証符号 ( *MAC* ) の各 *MAC* 生成対象データ領域の一部を共通データとし、いずれかの *MAC* 更新時に、前記共通データを更新し、他方の *MAC* 更新処理も併せて実行する構成としたことを特徴とするデータ処理装置にある。

## 【 0 0 2 2 】

さらに、本発明の第3の側面は、

記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行するデータ処理方法であり、

前記検証値を、コンテンツのカテゴリ毎に独立した検証値として生成して格納することを特徴とするデータ処理方法にある。

【0023】

さらに、本発明のデータ処理方法の一実施態様において、コンテンツ利用に際して、利用対象コンテンツ構成データに基づいて検証値を算出し、該算出検証値と、予め格納された検証値との比較処理を実行し、一致する場合にのみコンテンツ利用を行なうことを特徴とする。

【0024】

さらに、本発明のデータ処理方法の一実施態様において、前記記憶装置は、複数のディレクトリの各々に異なるカテゴリのコンテンツデータを格納する構成を有し、前記検証値は、前記複数のディレクトリの各々を単位とするコンテンツの集合に対して生成することを特徴とする。

【0025】

さらに、本発明のデータ処理方法の一実施態様において、前記記憶装置は、フラッシュメモリであり、前記カテゴリ毎の検証値は、フラッシュメモリの使用禁止ブロックとして設定された領域に格納することを特徴とする。

【0026】

さらに、本発明のデータ処理方法の一実施態様において、前記カテゴリは、コンテンツの種類に基づいて設定され、コンテンツの種類別に独立した検証値を設定し、格納することを特徴とする。

【0027】

さらに、本発明のデータ処理方法の一実施態様において、前記カテゴリは、コンテンツの暗号処理鍵として設定されるコンテンツキーK<sub>con</sub>を暗号化して提供する有効化キープロック（EKB）の管理エンティティに基づいて設定され、有効化キープロック（EKB）の管理エンティティ別に独立した検証値を設定し

、格納することを特徴とする。

【0028】

さらに、本発明のデータ処理方法の一実施態様において、前記検証値は、検証対象となるコンテンツおよびヘッダ情報等のコンテンツ関連データを構成する部分データをメッセージとしてDES暗号処理によって生成されるメッセージ認証符号(MAC)に基づいて生成されるデータであることを特徴とする。

【0029】

さらに、本発明のデータ処理方法の一実施態様において、前記検証対象となるコンテンツおよびヘッダ情報には、異なるデータ領域をMAC生成対象とした複数のメッセージ認証符号(MAC)を含み、前記複数のメッセージ認証符号(MAC)は、各々のMAC生成対象データ領域の一部を共通とし、いずれかのMAC更新時に他方のMAC更新処理も併せて実行することを特徴とする。

【0030】

さらに、本発明の第4の側面は、

記憶装置に格納されるコンテンツまたはヘッダ情報中の改竄有無検証データとしてのメッセージ認証符号(MAC)を生成して格納するデータ処理方法であり

コンテンツまたはヘッダ情報中の異なるデータ領域をMAC生成対象として複数のメッセージ認証符号(MAC)を生成するとともに、前記複数のメッセージ認証符号(MAC)の各MAC生成対象データ領域の一部を共通データとし、いずれかのMAC更新時に、前記共通データを更新し、他方のMAC更新処理も併せて実行することを特徴とするデータ処理方法にある。

【0031】

さらに、本発明の第5の側面は、

記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行するデータ処理をコンピュータ・システム上で実行せしめるコンピュータ・プログラムを提供するプログラム提供媒体であって、前記コンピュータ・プログラムは、

検証値を、コンテンツのカテゴリ毎に独立した検証値として生成して格納する

ステップを有することを特徴とするプログラム提供媒体にある。

【0032】

なお、本発明の第3の側面に係るプログラム提供媒体は、例えば、様々なプログラム・コードを実行可能な汎用コンピュータ・システムに対して、コンピュータ・プログラムをコンピュータ可読な形式で提供する媒体である。媒体は、CDやFD、MOなどの記録媒体、あるいは、ネットワークなどの伝送媒体など、その形態は特に限定されない。

【0033】

このようなプログラム提供媒体は、コンピュータ・システム上で所定のコンピュータ・プログラムの機能を実現するための、コンピュータ・プログラムと提供媒体との構造上又は機能上の協働的關係を定義したものである。換言すれば、該提供媒体を介してコンピュータ・プログラムをコンピュータ・システムにインストールすることによって、コンピュータ・システム上では協働的作用が発揮され、本発明の他の側面と同様の作用効果を得ることができるのである。

【0034】

本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

【0035】

【発明の実施の形態】

〔システム概要〕

図1に本発明のデータ処理システムの適用可能なコンテンツ配信システム例を示す。コンテンツ配信手段10は、データ処理手段20に対して、コンテンツあるいはコンテンツキー、その他、認証処理キー等のデータを暗号化して送信する。データ処理手段20では、受信した暗号化コンテンツ、あるいは暗号化コンテンツキー等を復号してコンテンツあるいはコンテンツキー等を取得して、画像データ、音声データの再生、あるいは各種プログラムを実行する。コンテンツの配信手段10とデータ処理手段20との間のデータ交換は、インターネット等のネットワークを介して、あるいはDVD、CD、その他の流通可能な記憶媒体を介して実行される。

## 【 0 0 3 6 】

データ処理手段 2 0 は、例えばフラッシュメモリ等の記憶手段を備えたメモリーカード等のデータ記憶手段 3 0 にデータを格納して保存する。データ記憶手段 3 0 には、暗号処理機能を有する記憶手段としての例えばメモリーカード（具体例としてはメモリスティック（Memory Stick: 商標））が含まれる。データ処理手段 2 0 からデータ記憶手段 3 0 に対するデータ格納処理、およびデータ記憶手段 3 0 からデータ処理手段に対するデータ移動の際には、相互認証処理、およびデータの暗号処理が実行され不正なデータコピーの防止が図られる。

## 【 0 0 3 7 】

なお、データ処理手段 2 0 に含まれる各機器間でのコンテンツデータの移動も可能であり、この際にも機器間の相互認証処理、データの暗号処理が実行される。

## 【 0 0 3 8 】

コンテンツ配信手段 1 0 としては、インターネット 1 1、衛星放送 1 2、電話回線 1 3、DVD、CD等のメディア 1 4 等があり、一方、データ処理手段 2 0 のデバイスとしては、パーソナルコンピュータ（PC）2 1、ポータブルデバイス（PD）2 2、携帯電話、PDA（Personal Digital Assistants）等の携帯機器 2 3、DVD、CDプレーヤ等の記録再生器、ゲーム端末 2 4、メモリーカード（ex. メモリスティック（商標））を利用した再生装置 2 5 等がある。これらデータ処理手段 2 0 の各デバイスは、コンテンツ配信手段 1 0 から提供されるコンテンツをネットワーク等の通信手段あるいは、他のデータ処理手段、または、データ記憶手段 3 0 から取得可能である。

## 【 0 0 3 9 】

図 2 に、代表的なコンテンツデータの移動処理例を示す。図 2 に示すシステムは、パーソナルコンピュータ（PC）1 0 0、再生装置 2 0 0 および記憶装置 3 0 0 間でのデータ（コンテンツ）の移動処理例を示した図である。PC 1 0 0 は、プログラムおよびデータ記憶用のハードディスク（HD）を有し、さらに、外部記憶媒体としてのCD、DVD等を装着可能な構成を持つ。

## 【 0 0 4 0 】

パーソナルコンピュータ（PC）100は、インターネット、公衆回線等の各種ネットワークに接続可能であり、例えば、EMD(Electronic Music Distribution: 電子音楽配信)などのサービスを提供する図示しないサービスプロバイダのホストコンピュータから、ネットワークをしてオーディオデータ、画像データ、プログラム等の各種データを受信し、受信したデータを必要に応じて復号して、再生装置200に出力する。また、パーソナルコンピュータ（PC）100は、コンテンツデータを受信するに当たって、必要に応じて、サービスプロバイダのホストコンピュータとの間で認証処理および課金処理などを行う。また、パーソナルコンピュータ（PC）100は、例えば、CD、DVDから入力したデータを再生装置200に出力する。

## 【0041】

記憶装置300は、再生装置200に対して着脱自在な装置、例えばメモリスティック(Memory Stick:商標)であり、フラッシュメモリなどの書き換え可能な半導体メモリを内蔵している。

## 【0042】

図2に示すように、PC100、再生装置200、記憶装置300間におけるデータ移動、例えば音楽データ、画像データ等のデータ再生、データ記録、データコピー等の処理の際にはデータ移動機器間において、相互認証処理が実行され、不正な機器を用いたデータ移動は防止される。これらの処理については後述する。また、コンテンツデータのネットワークまたは各種記憶媒体を介する配信、また、PCと再生装置相互間、あるいは再生装置とメモリカード等の記憶装置間でのコンテンツ移動の際にはコンテンツを暗号化することでデータのセキュリティが保全される。

## 【0043】

## [キー配信構成としてのツリー（木）構造について]

上述のようなコンテンツに対する暗号処理に適用する暗号鍵、例えばコンテンツの暗号処理に適用するコンテンツキー、またはコンテンツキーを暗号化するためのコンテンツキー暗号化キー等の様々な暗号処理キーを、安全に正当なライセンスを持つデバイスに配信する構成として、階層キー・ツリー構成について図3

以下を用いて説明する。

【0044】

図3の最下段に示すナンバ0～15がコンテンツデータの再生、実行を行なうデータ処理手段20を構成する個々のデバイス、例えばコンテンツ（音楽データ）再生装置である。すなわち図3に示す階層ツリー（木）構造の各葉（リーフ：leaf）がそれぞれのデバイスに相当する。

【0045】

各デバイス0～15は、製造時あるいは出荷時、あるいはその後において、図3に示す階層ツリー（木）構造における、自分のリーフからルートに至るまでのノードに割り当てられた鍵（ノードキー）および各リーフのリーフキーからなるキーセットをメモリに格納する。図3の最下段に示すK0000～K1111が各デバイス0～15にそれぞれ割り当てられたリーフキーであり、最上段のKR（ルートキー）から、最下段から2番目の節（ノード）に記載されたキー：KR～K111をノードキーとする。

【0046】

図3に示すツリー構成において、例えばデバイス0はリーフキーK0000と、ノードキー：K000、K00、K0、KRを所有する。デバイス5はK0101、K010、K01、K0、KRを所有する。デバイス15は、K1111、K111、K11、K1、KRを所有する。なお、図3のツリーにはデバイスが0～15の16個のみ記載され、ツリー構造も4段構成の均衡のとれた左右対称構成として示しているが、さらに多くのデバイスがツリー中に構成され、また、ツリーの各部において異なる段数構成を持つことが可能である。

【0047】

また、図3のツリー構造に含まれる各デバイスには、様々な記録媒体、例えば、デバイス埋め込み型あるいはデバイスに着脱自在に構成されたフラッシュメモリ等を使用したメモリカード、DVD、CD、MD等、様々なタイプの記憶装置を利用可能なデバイスが含まれている。さらに、様々なアプリケーションサービスが共存可能である。このような異なるデバイス、異なるアプリケーションの共存構成の上に図3に示すコンテンツあるいは鍵配布構成である階層ツリー構造が

適用される。

【 0 0 4 8 】

これらの様々なデバイス、アプリケーションが共存するシステムにおいて、例えば図3の点線で囲んだ部分、すなわちデバイス0, 1, 2, 3を同一の記録媒体を用いる1つのグループとして設定する。例えば、この点線で囲んだグループ内に含まれるデバイスに対しては、まとめて、共通のコンテンツを暗号化してプロバイダから送付したり、各デバイス共通に使用するコンテンツキーを送付したり、あるいは各デバイスからプロバイダあるいは決済機関等にコンテンツ料金の支払データをやはり暗号化して出力するといった処理が実行される。コンテンツプロバイダ、あるいは決済処理機関等、各デバイスとのデータ送受信を行なう機関は、図3の点線で囲んだ部分、すなわちデバイス0, 1, 2, 3を1つのグループとして一括してデータを送付する処理を実行する。このようなグループは、図3のツリー中に複数存在する。コンテンツプロバイダ、あるいは決済処理機関等、各デバイスとのデータ送受信を行なう機関は、メッセージデータ配信手段として機能する。

【 0 0 4 9 】

なお、ノードキー、リーフキーは、ある1つの鍵管理センタによって統括して管理してもよいし、各グループに対する様々なデータ送受信を行なうプロバイダ、決済機関等のメッセージデータ配信手段によってグループごとに管理する構成としてもよい。これらのノードキー、リーフキーは例えばキーの漏洩等の場合に更新処理が実行され、この更新処理は鍵管理センタ、プロバイダ、決済機関等が実行する。

【 0 0 5 0 】

このツリー構造において、図3から明らかなように、1つのグループに含まれる3つのデバイス0, 1, 2, 3はノードキーとして共通のキーK00、K0、KRを保有する。このノードキー共有構成を利用することにより、例えば共通のコンテンツキーをデバイス0, 1, 2, 3のみに提供することが可能となる。たとえば、共通に保有するノードキーK00自体をコンテンツキーとして設定すれば、新たな鍵送付を実行することなくデバイス0, 1, 2, 3のみに共通のコン

テンツキーの設定が可能である。また、新たなコンテンツキー  $K_{con}$  をノードキー  $K_{00}$  で暗号化した値  $Enc(K_{00}, K_{con})$  を、ネットワークを介してあるいは記録媒体に格納してデバイス 0, 1, 2, 3 に配布すれば、デバイス 0, 1, 2, 3 のみが、それぞれのデバイスにおいて保有する共有ノードキー  $K_{00}$  を用いて暗号  $Enc(K_{00}, K_{con})$  を解いてコンテンツキー:  $K_{con}$  を得ることが可能となる。なお、 $Enc(K_a, K_b)$  は  $K_b$  を  $K_a$  によって暗号化したデータであることを示す。

## 【0051】

また、ある時点  $t$  において、デバイス 3 の所有する鍵:  $K_{0011}, K_{001}, K_{00}, K_0, K_R$  が攻撃者（ハッカー）により解析されて露呈したことが発覚した場合、それ以降、システム（デバイス 0, 1, 2, 3 のグループ）で送受信されるデータを守るために、デバイス 3 をシステムから切り離す必要がある。そのためには、ノードキー:  $K_{001}, K_{00}, K_0, K_R$  をそれぞれ新たな鍵  $K(t)_{001}, K(t)_{00}, K(t)_0, K(t)_R$  に更新し、デバイス 0, 1, 2 にその更新キーを伝える必要がある。ここで、 $K(t)_{aaa}$  は、鍵  $K_{aaa}$  の世代 (Generation) :  $t$  の更新キーであることを示す。

## 【0052】

更新キーの配布処理について説明する。キーの更新は、例えば、図 4 (A) に示す有効化キーブロック (EKB: Enabling Key Block) と呼ばれるブロックデータによって構成されるテーブルをたとえばネットワーク、あるいは記録媒体に格納してデバイス 0, 1, 2 に供給することによって実行される。なお、有効化キーブロック (EKB) は、図 3 に示すようなツリー構造を構成する各リーフに対応するデバイスに新たに更新されたキーを配布するための暗号化キーによって構成される。有効化キーブロック (EKB) は、キー更新ブロック (KRB: Key Renewal Block) と呼ばれることもある。

## 【0053】

図 4 (A) に示す有効化キーブロック (EKB) には、ノードキーの更新の必要なデバイスのみが更新可能なデータ構成を持つブロックデータとして構成される。図 4 の例は、図 3 に示すツリー構造中のデバイス 0, 1, 2 において、世代

t の更新ノードキーを配布することを目的として形成されたブロックデータである。図 3 から明らかなように、デバイス 0, デバイス 1 は、更新ノードキーとして  $K(t)00$ 、 $K(t)0$ 、 $K(t)R$  が必要であり、デバイス 2 は、更新ノードキーとして  $K(t)001$ 、 $K(t)00$ 、 $K(t)0$ 、 $K(t)R$  が必要である。

## 【0054】

図 4 (A) の EKB に示されるように EKB には複数の暗号化キーが含まれる。最下段の暗号化キーは、 $Enc(K0010, K(t)001)$  である。これはデバイス 2 の持つリーフキー  $K0010$  によって暗号化された更新ノードキー  $K(t)001$  であり、デバイス 2 は、自身の持つリーフキーによってこの暗号化キーを復号し、 $K(t)001$  を得ることができる。また、復号により得た  $K(t)001$  を用いて、図 4 (A) の下から 2 段目の暗号化キー  $Enc(K(t)001, K(t)00)$  を復号可能となり、更新ノードキー  $K(t)00$  を得ることができる。以下順次、図 4 (A) の上から 2 段目の暗号化キー  $Enc(K(t)00, K(t)0)$  を復号し、更新ノードキー  $K(t)0$ 、図 4 (A) の上から 1 段目の暗号化キー  $Enc(K(t)0, K(t)R)$  を復号し  $K(t)R$  を得る。一方、デバイス  $K0000$ 、 $K0001$  は、ノードキー  $K000$  は更新する対象に含まれておらず、更新ノードキーとして必要なのは、 $K(t)00$ 、 $K(t)0$ 、 $K(t)R$  である。デバイス  $K0000$ 、 $K0001$  は、図 4 (A) の上から 3 段目の暗号化キー  $Enc(K000, K(t)00)$  を復号し  $K(t)00$  を取得し、以下、図 4 (A) の上から 2 段目の暗号化キー  $Enc(K(t)00, K(t)0)$  を復号し、更新ノードキー  $K(t)0$ 、図 4 (A) の上から 1 段目の暗号化キー  $Enc(K(t)0, K(t)R)$  を復号し  $K(t)R$  を得る。このようにして、デバイス 0, 1, 2 は更新した鍵  $K(t)001$ 、 $K(t)00$ 、 $K(t)0$ 、 $K(t)R$  を得ることができる。なお、図 4 (A) のインデックスは、復号キーとして使用するノードキー、リーフキーの絶対番地を示す。

## 【0055】

図 3 に示すツリー構造の上位段のノードキー： $K(t)0$ 、 $K(t)R$  の更新

が不要であり、ノードキーK 0 0のみの更新処理が必要である場合には、図4 (B)の有効化キープブロック(EKB)を用いることで、更新ノードキーK (t) 0 0をデバイス0, 1, 2に配布することができる。

## 【0056】

図4 (B)に示すEKBは、例えば特定のグループにおいて共有する新たなコンテンツキーを配布する場合に利用可能である。具体例として、図3に点線で示すグループ内のデバイス0, 1, 2, 3がある記録媒体を用いており、新たな共通のコンテンツキーK (t) c o nが必要であるとする。このとき、デバイス0, 1, 2, 3の共通のノードキーK 0 0を更新したK (t) 0 0を用いて新たな共通の更新コンテンツキー: K (t) c o nを暗号化したデータE n c (K (t), K (t) c o n)を図4 (B)に示すEKBとともに配布する。この配布により、デバイス4など、その他のグループの機器においては復号されないデータとしての配布が可能となる。

## 【0057】

すなわち、デバイス0, 1, 2はEKBを処理して得たK (t) 0 0を用いて上記暗号文を復号すれば、t時点でのコンテンツキーK (t) c o nを得ることが可能になる。

## 【0058】

## [EKBを使用したコンテンツキーの配布]

図5に、t時点でのコンテンツキーK (t) c o nを得る処理例として、K (t) 0 0を用いて新たな共通のコンテンツキーK (t) c o nを暗号化したデータE n c (K (t) 0 0, K (t) c o n)と図4 (B)に示すEKBとを記録媒体を介して受領したデバイス0の処理を示す。すなわちEKBによる暗号化メッセージデータをコンテンツキーK (t) c o nとした例である。

## 【0059】

図5に示すように、デバイス0は、記録媒体に格納されている世代: t時点のEKBと自分があらかじめ格納しているノードキーK 0 0 0を用いて上述したと同様のEKB処理により、ノードキーK (t) 0 0を生成する。さらに、復号した更新ノードキーK (t) 0 0を用いて更新コンテンツキーK (t) c o nを復

号して、後にそれを使用するために自分だけが持つリーフキー K 0 0 0 0 で暗号化して格納する。

#### 【0 0 6 0】

##### 〔E K B のフォーマット〕

図 6 に有効化キープブロック (E K B) のフォーマット例を示す。バージョン 6 0 1 は、有効化キープブロック (E K B) のバージョンを示す識別子である。なお、バージョンは最新の E K B を識別する機能とコンテンツとの対応関係を示す機能を持つ。デプスは、有効化キープブロック (E K B) の配布先のデバイスに対する階層ツリーの階層数を示す。データポインタ 6 0 3 は、有効化キープブロック (E K B) 中のデータ部の位置を示すポインタであり、タグポインタ 6 0 4 はタグ部の位置、署名ポインタ 6 0 5 は署名の位置を示すポインタである。

#### 【0 0 6 1】

データ部 6 0 6 は、例えば更新するノードキーを暗号化したデータを格納する。例えば図 5 に示すような更新されたノードキーに関する各暗号化キー等を格納する。

#### 【0 0 6 2】

タグ部 6 0 7 は、データ部に格納された暗号化されたノードキー、リーフキーの位置関係を示すタグである。このタグの付与ルールを図 7 を用いて説明する。図 7 では、データとして先に図 4 (A) で説明した有効化キープブロック (E K B) を送付する例を示している。この時のデータは、図 7 の表 (b) に示すようになる。このときの暗号化キーに含まれるトップノードのアドレスをトップノードアドレスとする。この場合は、ルートキーの更新キー K (t) R が含まれているので、トップノードアドレスは K R となる。このとき、例えば最上段のデータ E n c (K (t) 0, K (t) R) は、図 7 の (a) に示す階層ツリーに示す位置にある。ここで、次のデータは、E n c (K (t) 0 0, K (t) 0) であり、ツリー上では前のデータの左下の位置にある。データがある場合は、タグが 0、ない場合は 1 が設定される。タグは {左 (L) タグ, 右 (R) タグ} として設定される。最上段のデータ E n c (K (t) 0, K (t) R) の左にはデータがあるので、L タグ = 0、右にはデータがないので、R タグ = 1 となる。以下、すべ

てのデータにタグが設定され、図 7 (c) に示すデータ列、およびタグ列が構成される。

#### 【0063】

タグは、データ  $Enc(K_{xxx}, K_{yyy})$  がツリー構造のどこに位置しているのかを示すために設定されるものである。データ部に格納されるキーデータ  $Enc(K_{xxx}, K_{yyy}) \dots$  は、単純に暗号化されたキーの羅列データに過ぎないので、上述したタグによってデータとして格納された暗号化キーのツリー上の位置を判別可能としたものである。上述したタグを用いずに、先の図 4 で説明した構成のように暗号化データに対応させたノード・インデックスを用いて、例えば、

0 :  $Enc(K(t)_0, K(t)_{root})$   
 00 :  $Enc(K(t)_{00}, K(t)_0)$   
 000 :  $Enc(K((t)_{000}, K(T)_{00})$   
 . . . .

のようなデータ構成とすることも可能であるが、このようなインデックスを用いた構成とすると冗長なデータとなりデータ量が増大し、ネットワークを介する配信等においては好ましくない。これに対し、上述したタグをキー位置を示す索引データとして用いることにより、少ないデータ量でキー位置の判別が可能となる。

#### 【0064】

図 6 に戻って、EKB フォーマットについてさらに説明する。署名 (Signature) は、有効化キーブロック (EKB) を発行した例えば鍵管理センタ、コンテンツロバイダ、決済機関等が実行する電子署名である。EKB を受領したデバイスは署名検証によって正当な有効化キーブロック (EKB) 発行者が発行した有効化キーブロック (EKB) であることを確認する。

#### 【0065】

[EKB を使用したコンテンツキーおよびコンテンツの配信]

上述の例では、コンテンツキーのみを EKB とともに送付する例について説明したが、コンテンツキーで暗号化したコンテンツと、コンテンツキー暗号キーで

暗号化したコンテンツキーと、EKBによって暗号化したコンテンツキー暗号キーを併せて送付する構成について以下説明する。

## 【0066】

図8にこのデータ構成を示す。図8(a)に示す構成において、Enc(Kcon, content) 801は、コンテンツ(Content)をコンテンツキー(Kcon)で暗号化したデータであり、Enc(KEK, Kcon) 802は、コンテンツキー(Kcon)をコンテンツキー暗号キー(KEK: Key Encryption Key)で暗号化したデータであり、Enc(EKB, KEK) 803は、コンテンツキー暗号キーKEKを有効化キーブロック(EKB)によって暗号化したデータであることを示す。

## 【0067】

ここで、コンテンツキー暗号キーKEKは、図3で示すノードキー(K000, K00...)、あるいはルートキー(KR)自体であってもよく、またノードキー(K000, K00...)、あるいはルートキー(KR)によって暗号化されたキーであってもよい。

## 【0068】

図8(b)は、複数のコンテンツがメディアに記録され、それぞれが同じEnc(EKB, KEK) 805を利用している場合の構成例を示す、このような構成においては、各データに同じEnc(EKB, KEK)を付加することなく、Enc(EKB, KEK)にリンクするリンク先を示すデータを各データに付加する構成とすることができる。

## 【0069】

図9にコンテンツキー暗号キーKEKを、図3に示すノードキーK00を更新した更新ノードキーK(t)00として構成した場合の例を示す。この場合、図3の点線枠で囲んだグループにおいてデバイス3が、例えば鍵の漏洩によりリボーク(排除)されているとして、他のグループのメンバ、すなわち、デバイス0, 1, 2に対して図9に示す(a)有効化キーブロック(EKB)と、(b)コンテンツキー(Kcon)をコンテンツキー暗号キー(KEK=K(t)00)で暗号化したデータと、(c)コンテンツ(content)をコンテンツキー(Kc

on) で暗号化したデータとを配信することにより、デバイス 0, 1, 2 はコンテンツを得ることができる。

#### 【0070】

図9の右側には、デバイス0における復号手順を示してある。デバイス0は、まず、受領した有効化キーブロックから自身の保有するリーフキー $K_{000}$ を用いた復号処理により、コンテンツキー暗号キー( $KEK = K(t)_{00}$ )を取得する。次に、 $K(t)_{00}$ による復号によりコンテンツキー $K_{con}$ を取得し、さらにコンテンツキー $K_{con}$ によりコンテンツの復号を行なう。これらの処理により、デバイス0はコンテンツを利用可能となる。デバイス1, 2においても各々異なる処理手順で $EKB$ を処理することにより、コンテンツキー暗号キー( $KEK = K(t)_{00}$ )を取得することが可能となり、同様にコンテンツを利用することが可能となる。

#### 【0071】

図3に示す他のグループのデバイス4, 5, 6…は、この同様のデータ( $EKB$ )を受信したとしても、自身の保有するリーフキー、ノードキーを用いてコンテンツキー暗号キー( $KEK = K(t)_{00}$ )を取得することができない。同様にリボークされたデバイス3においても、自身の保有するリーフキー、ノードキーでは、コンテンツキー暗号キー( $KEK = K(t)_{00}$ )を取得することができず、正当な権利を有するデバイスのみがコンテンツを復号して利用することが可能となる。

#### 【0072】

このように、 $EKB$ を利用したコンテンツキーの配送を用いれば、データ量を少なくして、かつ安全に正当権利者のみが復号可能とした暗号化コンテンツを配信することが可能となる。

#### 【0073】

なお、有効化キーブロック( $EKB$ )、コンテンツキー、暗号化コンテンツ等は、ネットワークを介して安全に配信することが可能な構成であるが、有効化キーブロック( $EKB$ )、コンテンツキー、暗号化コンテンツをDVD、CD等の記録媒体に格納してユーザに提供することも可能である。この場合、記録媒体に

格納された暗号化コンテンツの復号には、同一の記録媒体に格納された有効化キーブロック（EKB）の復号により得られるコンテンツキーを使用するように構成すれば、予め正当権利者のみが保有するリーフキー、ノードキーによってのみ利用可能な暗号化コンテンツの配布処理、すなわち利用可能なユーザデバイスを限定したコンテンツ配布が簡易な構成で実現可能となる。

## 【0074】

図10に記録媒体に暗号化コンテンツとともに有効化キーブロック（EKB）を格納した構成例を示す。図10に示す例においては、記録媒体にコンテンツC1～C4が格納され、さらに各格納コンテンツに対応するの有効化キーブロック（EKB）を対応付けたデータが格納され、さらにバージョンMの有効化キーブロック（EKB\_\_M）が格納されている。例えばEKB\_\_1はコンテンツC1を暗号化したコンテンツキーKcon1を生成するのに使用され、例えばEKB\_\_2はコンテンツC2を暗号化したコンテンツキーKcon2を生成するのに使用される。この例では、バージョンMの有効化キーブロック（EKB\_\_M）が記録媒体に格納されており、コンテンツC3、C4は有効化キーブロック（EKB\_\_M）に対応付けられているので、有効化キーブロック（EKB\_\_M）の復号によりコンテンツC3、C4のコンテンツキーを取得することができる。EKB\_\_1、EKB\_\_2はディスクに格納されていないので、新たな提供手段、例えばネットワーク配信、あるいは記録媒体による配信によってそれぞれのコンテンツキーを復号するために必要なEKB\_\_1、EKB\_\_2を取得することが必要となる。

## 【0075】

## [階層ツリー構造のカテゴリー分類]

暗号鍵をルートキー、ノードキー、リーフキー等、図3の階層ツリー構造として構成し、コンテンツキー、認証キー、ICV生成キー、あるいはプログラムコード、データ等を有効化キーブロック（EKB）とともに暗号化して配信する構成について説明してきたが、ノードキー等を定義している階層ツリー構造を各デバイスのカテゴリー毎に分類して効率的なキー更新処理、暗号化キー配信、データ配信を実行する構成について、以下説明する。

## 【0076】

図 1 1 に階層ツリー構造のカテゴリの分類の一例を示す。図 1 1 において、階層ツリー構造の最上段には、ルートキー `K r o o t 1 1 0 1` が設定され、以下の中間段にはノードキー `1 1 0 2` が設定され、最下段には、リーフキー `1 1 0 3` が設定される。各デバイスは個々のリーフキーと、リーフキーからルートキーに至る一連のノードキー、ルートキーを保有する。

#### 【 0 0 7 7 】

ここで、一例として最上段から第 M 段目のあるノードをカテゴリノード `1 1 0 4` として設定する。すなわち第 M 段目のノードの各々を特定カテゴリのデバイス設定ノードとする。第 M 段の 1 つのノードを頂点として以下、M + 1 段以下のノード、リーフは、そのカテゴリに含まれるデバイスに関するノードおよびリーフとする。

#### 【 0 0 7 8 】

例えば図 1 1 の第 M 段目の 1 つのノード `1 1 0 5` にはカテゴリ「メモリスティック（商標）」が設定され、このノード以下に連なるノード、リーフはメモリスティックを使用した様々なデバイスを含むカテゴリ専用のノードまたはリーフとして設定される。すなわち、ノード `1 1 0 5` 以下を、メモリスティックのカテゴリに定義されるデバイスの関連ノード、およびリーフの集合として定義する。

#### 【 0 0 7 9 】

さらに、M 段から数段分下位の段をサブカテゴリノード `1 1 0 6` として設定することができる。例えば図に示すようにカテゴリ「メモリスティック」ノード `1 1 0 5` の 2 段下のノードに、メモリスティックを使用したデバイスのカテゴリに含まれるサブカテゴリノードとして、「再生専用器」のノードを設定する。さらに、サブカテゴリノードである再生専用器のノード `1 1 0 6` 以下に、再生専用器のカテゴリに含まれる音楽再生機能付き電話のノード `1 1 0 7` が設定され、さらにその下位に、音楽再生機能付き電話のカテゴリに含まれる「PHS」ノード `1 1 0 8` と「携帯電話」ノード `1 1 0 9` を設定することができる。

#### 【 0 0 8 0 】

さらに、カテゴリ、サブカテゴリは、デバイスの種類のみならず、例えばあるメーカー、コンテンツプロバイダ、決済機関等が独自に管理するノード、すなわ

ち処理単位、管轄単位、あるいは提供サービス単位等、任意の単位（これらを総称して以下、エンティティと呼ぶ）で設定することが可能である。例えば1つのカテゴリノードをゲーム機器メーカーの販売するゲーム機器XYZ専用の頂点ノードとして設定すれば、メーカーの販売するゲーム機器XYZにその頂点ノード以下の下段のノードキー、リーフキーを格納して販売することが可能となり、その後、暗号化コンテンツの配信、あるいは各種キーの配信、更新処理を、その頂点ノードキー以下のノードキー、リーフキーによって構成される有効化キーブロック（EKB）を生成して配信し、頂点ノード以下のデバイスに対してのみ利用可能なデータが配信可能となる。

#### 【0081】

このように、1つのノードを頂点として、以下のノードをその頂点ノードに定義されたカテゴリ、あるいはサブカテゴリの関連ノードとして設定する構成とすることにより、カテゴリ段、あるいはサブカテゴリ段の1つの頂点ノードを管理するメーカー、コンテンツプロバイダ等がそのノードを頂点とする有効化キーブロック（EKB）を独自に生成して、頂点ノード以下に属するデバイスに配信する構成が可能となり、頂点ノードに属さない他のカテゴリのノードに属するデバイスには全く影響を及ぼさずにキー更新を実行することができる。

#### 【0082】

##### [簡略EKBによるキー配信構成]

先に説明した例えば図3のツリー構成において、キー、例えばコンテンツキーを所定デバイス（リーフ）宛に送付する場合、キー配布先デバイスの所有しているリーフキー、ノードキーを用いて復号可能な有効化キーブロック（EKB）を生成して提供する。例えば図12（a）に示すツリー構成において、リーフを構成するデバイスa, g, jに対してキー、例えばコンテンツキーを送信する場合、a, g, jの各ノードにおいて復号可能な有効化キーブロック（EKB）を生成して配信する。

#### 【0083】

例えば更新ルートキーK(t) rootでコンテンツキーK(t) conを暗号化処理し、EKBとともに配信する場合を考える。この場合、デバイスa, g

、j は、それぞれが図 1 2 (b) に示すリーフおよびノードキーを用いて、E K B の処理を実行して K (t) r o o t を取得し、取得した更新ルートキー K (t) r o o t によってコンテンツキー K (t) c o n の復号処理を実行してコンテンツキーを得る。

## 【0084】

この場合に提供される有効化キーブロック (E K B) の構成は、図 1 3 に示すようになる。図 1 3 に示す有効化キーブロック (E K B) は、先の図 6 で説明した有効化キーブロック (E K B) のフォーマットにしたがって構成されたものであり、データ (暗号化キー) と対応するタグとを持つ。タグは、先に図 7 を用いて説明したように左 (L)、右 (R)、それぞれの方向にデータがあれば 0、無ければ 1 を示している。

## 【0085】

有効化キーブロック (E K B) を受領したデバイスは、有効化キーブロック (E K B) の暗号化キーとタグに基づいて、順次暗号化キーの復号処理を実行して上位ノードの更新キーを取得していく。図 1 3 に示すように、有効化キーブロック (E K B) は、ルートからリーフまでの段数 (デプス) が多いほど、そのデータ量は増加していく。段数 (デプス) は、デバイス (リーフ) 数に応じて増大するものであり、キーの配信先となるデバイス数が多い場合は、E K B のデータ量がさらに増大することになる。

## 【0086】

このような有効化キーブロック (E K B) のデータ量の削減を可能とした構成について説明する。図 1 4 は、有効化キーブロック (E K B) をキー配信デバイスに応じて簡略化して構成した例を示すものである。

## 【0087】

図 1 3 と同様、リーフを構成するデバイス a, g, j に対してキー、例えばコンテンツキーを送信する場合を想定する。図 1 4 の (a) に示すように、キー配信デバイスによってのみ構成されるツリーを構築する。この場合、図 1 2 (b) に示す構成に基づいて新たなツリー構成として図 1 4 (b) のツリー構成が構築される。K r o o t から K j までは全く分岐がなく 1 つの枝のみが存在すればよ

く、K r o o t から K a および K g に至るためには、K 0 に分岐点を構成するのみで、2 分岐構成の図 1 4 ( a ) のツリーが構築される。

#### 【 0 0 8 8 】

図 1 4 ( a ) に示すように、ノードとして K 0 のみを持つ簡略化したツリーが生成される。更新キー配信のための有効化キーブロック ( E K B ) は、これらの簡略ツリーに基づいて生成する。図 1 4 ( a ) に示すツリーは、有効化キーブロック ( E K B ) を復号可能な末端ノードまたはリーフを最下段とした 2 分岐型ツリーを構成するパスを選択して不要ノードを省略することにより再構築される再構築階層ツリーである。更新キー配信のための有効化キーブロック ( E K B ) は、この再構築階層ツリーのノードまたはリーフに対応するキーのみに基づいて構成される。

#### 【 0 0 8 9 】

先の図 1 3 で説明した有効化キーブロック ( E K B ) は、各リーフ a , g , j から K r o o t に至るまでのすべてのキーを暗号化したデータを格納していたが、簡略化 E K B は、簡略化したツリーを構成するノードについてのみの暗号化データを格納する。図 1 4 ( b ) に示すようにタグは 3 ビット構成を有する。第 1 および第 2 ビットは、図 1 3 の例と、同様の意味を持ち、左 ( L ) 、右 ( R ) 、それぞれの方向にデータがあれば 0 、無ければ 1 を示す。第 3 番目のビットは、E K B 内に暗号化キーが格納されているか否かを示すためのビットであり、データが格納されている場合は 1 、データが無い場合は、0 として設定される。

#### 【 0 0 9 0 】

データ通信網、あるいは記憶媒体に格納されてデバイス (リーフ) に提供される有効化キーブロック ( E K B ) は、図 1 4 ( b ) に示すように、図 1 3 に示す構成に比較すると、データ量が大幅に削減されたものとなる。図 1 4 に示す有効化キーブロック ( E K B ) を受領した各デバイスは、タグの第 3 ビットに 1 が格納された部分のデータのみを順次復号することにより、所定の暗号化キーの復号を実現することができる。例えばデバイス a は、暗号化データ E n c ( K a , K ( t ) 0 ) をリーフキー K a で復号して、ノードキー K ( t ) 0 を取得して、ノードキー K ( t ) 0 によって暗号化データ E n c ( K ( t ) 0 , K ( t ) r o o

t) を復号して  $K(t) \text{ root}$  を取得する。デバイス  $j$  は、暗号化データ  $Enc(K_j, K(t) \text{ root})$  をリーフキー  $K_j$  で復号して、 $K(t) \text{ root}$  を取得する。

#### 【0091】

このように、配信先のデバイスによってのみ構成される簡略化した新たなツリー構成を構築して、構築されたツリーを構成するリーフおよびノードのキーのみを用いて有効化キープロック (EKB) を生成することにより、少ないデータ量の有効化キープロック (EKB) を生成することが可能となり、有効化キープロック (EKB) のデータ配信が効率的に実行可能となる。

#### 【0092】

なお、簡略化した階層ツリー構成は、後段で説明するエンティティ単位の EKB 管理構成において特に有効に活用可能である。エンティティは、キー配信構成としてのツリー構成を構成するノードあるいはリーフから選択した複数のノードあるいはリーフの集合体ブロックである。エンティティは、デバイスの種類に応じて設定される集合であったり、あるいはデバイス提供メーカー、コンテンツプロバイダ、決済機関等の管理単位等、ある共通点を持った処理単位、管轄単位、あるいは提供サービス単位等、様々な態様の集合として設定される。1つのエンティティには、ある共通のカテゴリに分類されるデバイスが集まっており、例えば複数のエンティティの頂点ノード (サブルート) によって上述したと同様の簡略化したツリーを再構築して EKB を生成することにより、選択されたエンティティに属するデバイスにおいて復号可能な簡略化された有効化キープロック (EKB) の生成、配信が可能となる。エンティティ単位の管理構成については後段で詳細に説明する。

#### 【0093】

なお、このような有効化キープロック (EKB) は、光ディスク、DVD等の情報記録媒体に格納した構成とすることが可能である。例えば、上述の暗号化キーデータによって構成されるデータ部と、暗号化キーデータの階層ツリー構造における位置識別データとしてのタグ部とを含む有効化キープロック (EKB) にさらに、更新ノードキーによって暗号化したコンテンツ等のメッセージデータと

を格納した情報記録媒体を各デバイスに提供する構成が可能である。デバイスは有効化キープブロック（EKB）に含まれる暗号化キーデータをタグ部の識別データにしたがって順次抽出して復号し、コンテンツの復号に必要なキーを取得してコンテンツの利用を行なうことが可能となる。もちろん、有効化キープブロック（EKB）をインターネット等のネットワークを介して配信する構成としてもよい。

#### 【0094】

##### 〔暗号処理機能を有する記憶装置とデータ処理装置間のデータ移動〕

次に、上述した階層ツリー構成を適用した有効化キープブロック（EKB）によって配信される暗号処理キーを適用した処理構成について、暗号処理機能を有する記憶装置、例えばメモリスティック（商標）等のメモリカードと、データ再生装置間におけるデータ移動処理を中心として説明する。

#### 【0095】

図15は、相互にコンテンツデータの移動を実行可能な再生装置と暗号処理機能を有するメモリカード等の記憶装置の詳細構成を示したブロック図である。

#### 【0096】

図15に示すように、記憶装置300は、例えば、主制御モジュール31、通信インターフェイス32、制御モジュール33、フラッシュメモリ34およびフラッシュメモリ管理モジュール35を有する。以下、各モジュールについて説明する。

#### 【0097】

##### 〔制御モジュール33〕

図15に示すように、制御モジュール33は、例えば、乱数発生ユニット50、記憶ユニット51、鍵生成／演算ユニット52、相互認証ユニット53、暗号化／復号ユニット54および制御ユニット55を有する。制御モジュール33は、シングルチップの暗号処理専用の集積回路であり、多層構造を有し、内部のメモリセルはアルミニウム層などのダミー層に挟まれている。また、制御モジュール33は、動作電圧または動作周波数の幅が狭く、外部から不正にデータを読み出せないように耐タンパー性を有している。乱数発生ユニット50は、乱数発生

指示を受けると、64ビット（8バイト）の乱数を発生する。

#### 【0098】

記憶ユニット51は、例えば、EEPROM(Electrically Erasable Programmable Read Only Memory)などの不揮発性メモリであり、認証処理に必要な鍵データなどの種々のデータを記憶している。図16は、記憶ユニット51に記憶されているデータを説明するための図である。図16に示すように、記憶ユニット51は、認証鍵データIK0～IK31、装置識別データIDmおよび記憶用鍵データKst mを記憶している。

#### 【0099】

認証鍵データIK0～IK31は、記憶装置300が再生装置200との間で相互認証を行う際に用いられる鍵データであり、後述するように相互認証を行う度に認証鍵データIK0～IK31のうちの認証鍵データがランダムに選択される。なお、認証鍵データIK0～IK31および記憶用鍵データKst mは、記憶装置300の外部から読めないようになっている。装置識別データIDmは、記憶装置300に対してユニークに付けられた識別データであり、後述するように、記憶装置300が再生装置200との間で相互認証を行う際に読み出されて再生装置200に出力される。記憶用鍵データKst mは、後述するように、コンテンツの暗号化に用いられるコンテンツ鍵データCKを暗号化してフラッシュメモリ34に記憶する際に用いられる。

#### 【0100】

鍵生成／演算ユニット52は、例えば、ISO/IEC9797のMAC(Message Authentication Code)演算などの種々の演算を行って鍵データを生成する。このとき、MAC演算には、例えば、“Block cipher Algorithm”としてFIPS PUB 46-2に規定されるDES(Data Encryption Standard)が用いられる。MAC演算は、任意の長さのデータを固定の長さに圧縮する一方向性ハッシュ関数演算であり、関数値が秘密鍵に依存して定まる。

#### 【0101】

相互認証ユニット53は、再生装置200からオーディオデータを入力してフラッシュメモリ34に書き込む動作を行うのに先立って、再生装置200との間

で相互認証処理を行う。また、相互認証ユニット 5 3 は、フラッシュメモリ 3 4 からオーディオデータを読み出して再生装置 2 0 0 に出力する動作を行うのに先立って、再生装置 2 0 0 との間で相互認証処理を行う。また、相互認証ユニット 5 3 は、相互認証処理において、前述した MAC 演算を行う。当該相互認証処理では、記憶ユニット 5 1 に記憶されているデータが用いられる。

#### 【 0 1 0 2 】

暗号化／復号ユニット 5 4 は、DES、IDEA、MISTYなどのブロック暗号アルゴリズムでの暗号化を行う。使用するモードは、FIPS PUB 81 "DES MODES OF OPERATION" に規定されているような ECB (Electronic Code Book) モードおよび CBC (Cipher Block Chaining) モードである。また、暗号化／復号ユニット 5 4 は、DES、IDEA、MISTYなどのブロック復号アルゴリズムでの復号を行う。使用するモードは、上記 ECB モードおよび CBC モードである。当該 ECB モードおよび CBC モードのブロック暗号化／復号では、指定された鍵データを用いて指定されたデータを暗号化／復号する。制御ユニット 5 5 は、乱数発生ユニット 5 0、記憶ユニット 5 1、鍵生成／演算ユニット 5 2、相互認証ユニット 5 3 および暗号化／復号ユニット 5 4 の処理を統括して制御する。

#### 【 0 1 0 3 】

##### 〔フラッシュメモリ 3 4〕

フラッシュメモリ 3 4 は、例えば、3 2 M バイトの記憶容量を有する。フラッシュメモリ 3 4 には、相互認証ユニット 5 3 による再生装置 2 0 0 と記憶装置 3 0 0 との間の相互認証処理によって双方が正当な装置であると認められたときに、再生装置 2 0 0 から入力したオーディオデータあるいは画像データ等、各種データが書き込まれる。また、フラッシュメモリ 3 4 からは、相互認証ユニット 5 3 による再生装置 2 0 0 と記憶装置 3 0 0 との間の相互認証処理によって正当な相手であると認められたときに、オーディオデータ、画像データ等が読み出されて再生装置 2 0 0 に出力される。

#### 【 0 1 0 4 】

以下、フラッシュメモリ 3 4 に記憶されるデータおよびそのフォーマットにつ

いて説明する。図 1 7 は、フラッシュメモリ 3 4 に記憶されるデータを説明するための図である。図 1 7 に示すように、フラッシュメモリ 3 4 には、例えば、再生管理ファイル、複数のトラックデータ（再生データ）ファイルが記憶されている。ここで、再生管理ファイルはトラックデータファイルの再生を管理する管理データを有し、トラックデータファイルはそれぞれ対応するトラックデータ（オーディオデータ）を有している。なお、本実施形態では、トラックデータは、例えば、1 曲分のオーディオデータを意味する。以下、フラッシュメモリ 3 4 に記憶されるデータをオーディオデータとした場合の例について説明する。

#### 【0 1 0 5】

図 1 8 は、再生管理ファイルの構成を示し、図 1 9 が一つ（1 曲）の A T R A C 3 データファイルの構成を示す。再生管理ファイルは、1 6 K B 固定長のファイルである。A T R A C 3 データファイルは、曲単位でもって、先頭の属性ヘッダと、それに続く実際の暗号化された音楽データとからなる。属性ヘッダも 1 6 K B 固定長とされ、再生管理ファイルと類似した構成を有する。

#### 【0 1 0 6】

再生管理ファイルは、ヘッダ、1 バイトコードのメモリカードの名前 N M 1 - S、2 バイトコードのメモリカードの名前 N M 2 - S、曲順の再生テーブル T R K T B L、メモリカード全体の付加情報 I N F - S とからなる。データファイルの先頭の属性ヘッダは、ヘッダ、1 バイトコードの曲名 N M 1、2 バイトコードの曲名 N M 2、トラックのキー情報等のトラック情報 T R K I N F、パーツ情報 P R T I N F と、トラックの付加情報 I N F とからなる。ヘッダには、総パーツ数、名前の属性、付加情報のサイズの情報等が含まれる。

#### 【0 1 0 7】

属性ヘッダに対して A T R A C 3 の音楽データが続く。音楽データは、1 6 K B のブロック毎に区切られ、各ブロックの先頭にヘッダが付加されている。ヘッダには、暗号を復号するための初期値が含まれる。なお、暗号化の処理を受けるのは、A T R A C 3 データファイル中の音楽データ等のコンテンツデータのみであって、それ以外の再生管理ファイル、ヘッダ等のデータは、暗号化されない。

#### 【0 1 0 8】

図20は、再生管理ファイルPBLISTの詳細なデータ構成を示す。再生管理ファイルPBLISTは、1クラスタ（1ブロック＝16KB）のサイズである。図20Aに示すヘッダは、32バイトから成る。図20Bに示すヘッダ以外の部分は、メモ리카ード全体に対する名前NM1-S（256バイト）、名前NM2-S（512バイト）、暗号化されたコンテンツキー（CONTENTSKY）、MAC、S-YMDhmsと、再生順番を管理するテーブルTRKTB（800バイト）、メモ리카ード全体に対する付加情報INF-S（14720バイト）および最後にヘッダ中の情報の一部が再度記録されている。これらの異なる種類のデータ群のそれぞれの先頭は、再生管理ファイル内で所定の位置となるように規定されている。

## 【0109】

再生管理ファイルは、図20Aに示す（0x0000）および（0x0010）で表される先頭から32バイトがヘッダである。なお、ファイル中で先頭から16バイト単位で区切られた単位をスロットと称する。ファイルの第1および第2のスロットに配されるヘッダには、下記の意味、機能、値を持つデータが先頭から順に配される。なお、Reservedと表記されているデータは、未定義のデータを表している。通常ヌル（0x00）が書かれるが、何が書かれていてもReservedのデータが無視される。将来のバージョンでは、変更がありうる。また、この部分への書き込みは禁止する。Optionと書かれた部分も使用しない場合は、全てReservedと同じ扱いとされる。

## 【0110】

BLKID-TL0（4バイト）

意味：BLOCKID FILE ID

機能：再生管理ファイルの先頭であることを識別するための値

値：固定値＝"TL＝0"（例えば0x544C2D30）

MCODE（2バイト）

意味：MAKER CODE

機能：記録した機器の、メーカー、モデルを識別するコード

値：上位10ビット（メーカーコード） 下位6ビット（機種コード）

REVISION (4 バイト)

意味: PBLIST の書き換え回数

機能: 再生管理ファイルを書き換える度にインクリメント

値: 0 より始まり +1 ずつ増加する

【0111】

SN1C+L (2 バイト)

意味: NM1-S 領域に書かれるメモ리카ードの名前 (1 バイト) の属性を表す

機能: 使用する文字コードと言語コードを各 1 バイトで表す

値: 文字コード (C) は上位 1 バイトで下記のように文字を区別する

00: 文字コードは設定しない。単なる 2 進数として扱うこと

01: ASCII (American Standard Code for Information Interchange)

02: ASCII+KANA 03: modified8859-1

81: MS-JIS 82: KS C 5601-1989 83: GB (Great Britain) 2312-80

90: S-JIS (Japanese Industrial Standards) (for Voice)。

【0112】

言語コード (L) は下位 1 バイトで下記のように EBU Tech 3258 規定に準じて言語を区別する

00: 設定しない 08: German 09: English 0A: Spanish

0F: French 15: Italian 1D: Dutch

65: Korean 69: Japanese 75: Chinese

データが無い場合オールゼロとすること。

【0113】

SN2C+L (2 バイト)

意味: NM2-S 領域に書かれるメモ리카ードの名前 (2 バイト) の属性を表す

機能: 使用する文字コードと言語コードを各 1 バイトで表す

値: 上述した SN1C+L と同一

SINFSIZE (2 バイト)

意味：I N F - S 領域に書かれるメモリカード全体に関する付加情報の全てを合計したサイズを表す

機能：データサイズを16バイト単位の大きさを記述、無い場合は必ずオールゼロとすること

値：サイズは0x0001から0x39C(924)

T - T R K (2バイト)

意味：TOTAL TRACK NUMBER

機能：総トラック数

値：1から0x0190(最大400トラック)、データが無い場合はオールゼロとすること

V e r N o (2バイト)

意味：フォーマットのバージョン番号

機能：上位がメジャーバージョン番号、下位がマイナーバージョン番号。著作権対応型か否か、すなわち前述の階層ツリー構成による有効化キーブロック(EKB)による配信キーの使用対象か否かを示すデータとしても使用される。

値：例 0x0100 (V e r 1. 0)  
0x0203 (V e r 2. 3)。

【0114】

上述したヘッダに続く領域に書かれるデータ(図20B)について以下に説明する。

【0115】

N M 1 - S

意味：メモリカード全体に関する1バイトの名前

機能：1バイトの文字コードで表した可変長の名前データ(最大で256)名前データの終了は、必ず終端コード(0x00)を書き込むこと

サイズはこの終端コードから計算すること、データの無い場合は少なくとも先頭(0x0020)からヌル(0x00)を1バイト以上記録すること

値：各種文字コード

N M 2 - S

機能：2バイトの文字コードで表した可変長の名前データ（最大で512）名

サイズはこの終端コードから計算すること、データの無い場合は少なくとも先

値：各種文字コード。

EKB\_ v e r s i o n (4バイト)

機能：階層ツリー構成による有効化キープロック（EKB）によって提供されるコンテンツキーを求めるための有効化キープロック（EKB）を示す。

値：0 から 0 x FF まで

【0 1 1 7】

E (K s t m, K c o n) (8バイト)

意味：コンテンツ毎の暗号処理用のキーであるコンテンツキーをメモリカードのストレージキー（K s t m）で暗号化したデータ。

機能：コンテンツの暗号処理に使用される

値：0から0xFFFFFFFFFFFFFFFFFFFFFまで

【0 1 1 8】

E (KEK<sub>n</sub>, Kcon) (8バイト)

意味：コンテンツ毎の暗号処理用のキーであるコンテンツキーを前述の階層ツリー構成による有効化キーブロック（EKB）によって提供されるキー暗号化キーKEK<sub>n</sub>によって暗号化したデータ。

機能：コンテンツの暗号処理に使用される

値：0から0xFFFFFFFFFFFFFFFFFFFFFまで

【 0 1 1 9 】

C\_MAC [0] (8バイト)

意味：著作権情報改ざんチェック値

機能：再生管理ファイル内のデータ、最終コンテンツ記録等のコンテンツ処理日時を示す S-YMD h m s 他のデータに基づいて生成される改竄チェック用の値。日時データ S-YMD h m s が改竄されていた場合には、C\_MAC [0] のチェック時に改竄があったと判定され、コンテンツの再生が実行されない。

値：0から0xFFFFFFFFFFFFFFFFFFFFFまで。

**【 0 1 2 0 】**

M G R

意味：コンテンツキーの種類

機能：0x00で、コンテンツキーKconと、E (KEKn, Kcon) の両方有り、0x01で、E (KEKn, Kcon) のみ有り。

値：0から0 x 0 1まで

**【 0 1 2 1 】**

S - Y M D h m s (4 バイト) (O p t i o n)

意味：信頼できる時計を持つ機器で記録した年・月・日・時・分・秒

機能：コンテンツの最終記録日時等、コンテンツ最終処理日時を識別するための値。コンテンツの処理時に更新される。

値：25-31ビット 年 0-99 (1980-2079)

21-24ビット 月 0-12

16-20ビット 目 0-31

11-15ビット 時 0-23

05-10ビット 分 0-59

00-04ビット 秒 0-29 (2秒単位)。

なお、S-YMDhms は、コンテンツ記録時等のコンテンツ処理時に更新され、更新されたデータに基づいて前述のC-MAC〔0〕も更新されて格納される。

【0 1 2 2】

TRK - n n n

意味：再生する ATRAC3 データファイルの SQN（シーケンス）番号

機能：TRKINFの中のFN<sub>0</sub>を記述する

値：1から400 (0x190)

トラックが存在しない時はオールゼロとすること

INF-S

意味：メモ리카ード全体に関する付加情報データ（例えば写真、歌詞、解説等の情報）

機能：ヘッダを伴った可変長の付加情報データ

複数の異なる付加情報が並べられることがある。それぞれにIDとデータサイズが付けられている。個々のヘッダを含む付加情報データは最小16バイト以上で4バイトの整数倍の単位で構成される。その詳細については、後述する

値：付加情報データ構成を参照

【0123】

再生管理ファイルの最後のスロットとして、ヘッダ内のものと同一のBLKID-TL0と、MC<sub>0</sub>deと、REVISIONとが書かれる。

【0124】

民生用オーディオ機器として、メモ리카ードが記録中に抜かれたり、電源が切れることがあり、復活した時にこれらの異常の発生を検出することが必要とされる。上述したように、REVISIONをブロックの先頭と末尾に書き込み、この値を書き換える度に+1インクリメントするようにしている。若し、ブロックの途中で異常終了が発生すると、先頭と末尾のREVISIONの値が一致せず、異常終了を検出することができる。REVISIONが2個存在するので、高い確率で異常終了を検出することができる。異常終了の検出時には、エラーメッセージの表示等の警告が発生する。

【0125】

また、1ブロック（16KB）の先頭部分に固定値BLKID-TL0を挿入しているので、FATが壊れた場合の修復の目安に固定値を使用できる。すなわち、各ブロックの先頭の固定値を見れば、ファイルの種類を判別することが可能である。しかも、この固定値BLKID-TL0は、ブロックのヘッダおよびブロックの終端部分に二重に記述するので、その信頼性のチェックを行うことがで

きる。なお、再生管理ファイル P B L I S T の同一のものを二重に記録しても良い。

## 【 0 1 2 6 】

A T R A C 3 データファイルは、トラック情報管理ファイルと比較して、相当大きなデータ量であり、A T R A C 3 データファイルに関しては、ブロック番号 B L O C K S E R I A L が付けられている。但し、A T R A C 3 データファイルは、通常複数のファイルがメモリカード上に存在するので、C O N N U M 0 でコンテンツの区別を付けた上で、B L O C K S E R I A L を付けないと、重複が発生し、F A T が壊れた場合のファイルの復旧が困難となる。換言すると単一の A T R A C 3 データファイルは、複数の B L O C K で構成されると共に、離散して配置される可能性があるので、同一 A T R A C 3 データファイルを構成する B L O C K を判別するために C O N N U M 0 を用いると共に、同一 A T R A C 3 データファイル内の昇降順をブロック番号 B L O C K S E R I A L で決定する。

## 【 0 1 2 7 】

同様に、F A T の破壊までにはいたらないが、論理を間違ってファイルとして不都合のあるような場合に、書き込んだメーカーの機種が特定できるように、メーカーコード (M C o d e) がブロックの先頭と末尾に記録されている。

## 【 0 1 2 8 】

図 2 0 C は、付加情報データの構成を示す。付加情報の先頭に下記のヘッダが書かれる。ヘッダ以降に可変長のデータが書かれる。

## 【 0 1 2 9 】

I N F

意味 : F I E L D I D

機能 : 付加情報データの先頭を示す固定値

値 : 0 x 6 9

I D

意味 : 付加情報キーコード

機能 : 付加情報の分類を示す

値：0 から 0 x F F

## S I Z E

意味：個別の付加情報の大きさ

機能：データサイズは自由であるが、必ず4バイトの整数倍でなければならない。また、最小16バイト以上のこと。データの終わりより余りができる場合はヌル(0x00)で埋めておくこと

値：16 から 1 4 7 8 4 (0 x 3 9 C 0)

## M C o d e

意味：MAKER CODE

機能：記録した機器の、メーカー、モデルを識別するコード

値：上位10ビット(メーカーコード) 下位6ビット(機種コード)

## C + L

意味：先頭から12バイト目からのデータ領域に書かれる文字の属性を表す

機能：使用する文字コードと言語コードを各1バイトで表す

値：前述のSNC+Lと同じ

## D A T A

意味：個別の付加情報データ

機能：可変長データで表す。実データの先頭は常に12バイト目より始まり、長さ(サイズ)は最小4バイト以上、常に4バイトの整数倍でなければならない。データの最後から余りがある場合はヌル(0x00)で埋めること

値：内容により個別に定義される。

【0130】

図21に、ATRAC3データファイルA3Dnnnnのデータ配列例を示す。図21には、データファイルの属性ヘッダ(1ブロック)と、音楽データファイル(1ブロック)とが示されている。図21では、この2ブロック(16x2=32Kバイト)の各スロットの先頭のバイト(0x0000~0x7FF0)が示されている。図22に分離して示すように、属性ヘッダの先頭から32バイトがヘッダであり、256バイトが曲名領域NM1(256バイト)であり、512バイトが曲名領域NM2(512バイト)である。属性ヘッダのヘッダには

、下記のデータが書かれる。

【0131】

BLKID-HD0 (4バイト)

意味: BLOCKID FILE ID

機能: AT-RAC3データファイルの先頭であることを識別するための値

値: 固定値="HD=0" (例えば0x48442D30)

【0132】

MCODE (2バイト)

意味: MAKER CODE

機能: 記録した機器の、メーカー、モデルを識別するコード

値: 上位10ビット (メーカーコード) 下位6ビット (機種コード)

【0133】

BLOCK SERIAL (4バイト)

意味: トラック毎に付けられた連続番号

機能: ブロックの先頭は0から始まり次のブロックは+1ずつインクリメント

編集されても値を変化させない

値: 0より始まり0xFFFFFFFまで。

【0134】

NC+L (2バイト)

意味: トラック (曲名) データ (NM1) の属性を表す

機能: NM1に使用される文字コードと言語コードを各1バイトで表す

値: SN1C+Lと同一

【0135】

N2C+L (2バイト)

意味: トラック (曲名) データ (NM2) の属性を表す

機能: NM2に使用される文字コードと言語コードを各1バイトで表す

値: SN1C+Lと同一

【0136】

INFSIZE (2バイト)

意味：トラックに関する付加情報の全てを合計したサイズを表す

機能：データサイズを 1 6 バイト単位の大きさを記述、無い場合は必ずオールゼロとすること

値：サイズは 0 x 0 0 0 0 から 0 x 3 C 6 ( 9 6 6 )

【 0 1 3 7 】

T - P R T ( 2 バイト )

意味：トータルパーツ数

機能：トラックを構成するパーツ数を表す。通常は 1

値：1 から 0 x 2 8 5 ( 6 4 5 dec )

【 0 1 3 8 】

T - S U ( 4 バイト )

意味：トータル S U ( サウンドユニット ) 数、S U は、パーツの最小単位であり、且つ A T R A C 3 でオーディオデータを圧縮する時の最小のデータ単位である。4 4 . 1 k H z のサンプリング周波数で得られた 1 0 2 4 サンプル分 ( 1 0 2 4 x 1 6 ビット x 2 チャンネル ) のオーディオデータを約 1 / 1 0 に圧縮した数百バイトのデータが S U である。1 S U は、時間に換算して約 2 3 m 秒になる。通常は、数千に及ぶ S U によって 1 つのパーツが構成される。1 クラスタが 4 2 個の S U で構成される場合、1 クラスタで約 1 秒の音を表すことができる。1 つのトラックを構成するパーツの数は、付加情報サイズに影響される。パーツ数は、1 ブロックの中からヘッダや曲名、付加情報データ等を除いた数で決まるために、付加情報が全く無い状態が最大数 ( 6 4 5 個 ) のパーツを使用できる条件となる。

機能：1 トラック中の実際の総 S U 数を表す。曲の演奏時間に相当する

値：0 x 0 1 から 0 x 0 0 1 F F F F F

【 0 1 3 9 】

I N X ( 2 バイト ) ( O p t i o n )

意味：INDEX の相対場所

機能：曲のさびの部分 ( 特徴的な部分 ) の先頭を示すポインタ。曲の先頭からの位置を S U の個数を 1 / 4 した数で指定する。これは、通常の S U の 4 倍の長

さの時間（約 9 3 m 秒）に相当する

値：0 から 0 x F F F F（最大、約 6 0 8 4 秒）

【0 1 4 0】

X T（2 バイト）（O p t i o n）

意味：INDEX の再生時間

機能：I N X-nnn で指定された先頭から再生すべき時間の S U の個数を 1 / 4 した数で指定する。これは、通常の S U の 4 倍の長さの時間（約 9 3 m 秒）に相当する

値：0 x 0 0 0 0：無設定      0 x 0 1 から 0 x F F F E（最大 6 0 8 4 秒）

0 x F F F F：曲の終わりまで。

【0 1 4 1】

次に曲名領域 N M 1 および N M 2 について説明する。

【0 1 4 2】

N M 1

意味：曲名を表す文字列

機能：1 バイトの文字コードで表した可変長の曲名（最大で 2 5 6）

名前データの終了は、必ず終端コード（0 x 0 0）を書き込むこと

サイズはこの終端コードから計算すること、データの無い場合は少なくとも先頭（0 x 0 0 2 0）からヌル（0 x 0 0）を 1 バイト以上記録すること

値：各種文字コード

【0 1 4 3】

N M 2

意味：曲名を表す文字列

機能：2 バイトの文字コードで表した可変長の名前データ（最大で 5 1 2）

名前データの終了は、必ず終端コード（0 x 0 0）を書き込むこと

サイズはこの終端コードから計算すること、データの無い場合は少なくとも先頭（0 x 0 1 2 0）からヌル（0 x 0 0）を 2 バイト以上記録すること

値：各種文字コード。

【0 1 4 4】

属性ヘッダの固定位置 (0x320) から始まる、80バイトのデータをトラック情報領域 TRKINF と呼び、主としてセキュリティ関係、コピー制御関係の情報を一括して管理する。図 23 に TRKINF の部分を示す。TRKINF 内のデータについて、配置順序に従って以下に説明する。

## 【0145】

EKI (1バイト)

意味：前述の階層ツリー構成による有効化キーブロック (EKB) によって提供される暗号化コンテンツキー：E (KEKn, Kcon) を有するか否かを示す。

機能：bit 7 = 1 でキー有、bit 7 = 0 で無し。bit 7 = 0 の場合は、EKB\_\_version、E (KEKn, Kcon) は非参照。

値：0 から 0xFF まで

## 【0146】

EKB\_\_version (4バイト)

意味：前述の階層ツリー構成による有効化キーブロック (EKB) によって提供されるコンテンツキーの世代番号、および／または有効化キーブロック (EKB) のファイル名を示す。

機能：階層ツリー構成による有効化キーブロック (EKB) によって提供されるコンテンツキーを求めるための有効化キーブロック (EKB) を示す。

値：0 から 0xFF まで

## 【0147】

E (Kstm, Kcon) (8バイト)

意味：コンテンツ毎の暗号処理用のキーであるコンテンツキーをメモリカードのストレージキー (Kstm) で暗号化したデータ。

機能：コンテンツの暗号処理に使用される

値：0 から 0xFFFFFFFFFFFFFFFF まで

## 【0148】

E (KEKn, Kcon) (8バイト)

意味：コンテンツ毎の暗号処理用のキーであるコンテンツキーを前述の階層ツ

機能：コンテンツの暗号処理に使用される

【 0 1 4 9 】

意味：著作権情報改ざんチェック値

【0 1 5 0】

意味：パーツの属性

値：図 2 4 を参照して以下に説明する

【 0 1 5 1 】

4 6

Bのメモ리카ードの場合)、データ転送レート、1ブロック内のSU数がそれぞれ示されている。1SUのバイト数は、(モノ:136バイト、LP:192バイト、SP:304バイト、EX:384バイト、HQ:512バイト)である。さらに、ビット7によって、ATRAC3のモード(0:Dual 1:Joint)が示される。

【0152】

一例として、64MBのメモ리카ードを使用し、SPモードの場合について説明する。64MBのメモ리카ードには、3968ブロックがある。SPモードでは、1SUが304バイトであるので、1ブロックに53SUが存在する。1SUは、(1024/44100)秒に相当する。従って、1ブロックは、(1024/44100)×53×(3968-16)=4863秒=81分

転送レートは、

$$(44100/1024) \times 304 \times 8 = 104737 \text{ bps}$$

となる。

【0153】

LT (1バイト)

意味:再生制限フラグ(ビット7およびビット6)とセキュリティバージョン(ビット5-ビット0)

機能:このトラックに関して制限事項があることを表す

値:ビット7: 0=制限なし 1=制限有り

ビット6: 0=期限内 1=期限切れ

ビット5-ビット0:セキュリティバージョン0(0以外であれば再生禁止とする)

【0154】

FN0 (2バイト)

意味:ファイル番号

機能:最初に記録された時のトラック番号、且つこの値は、メモ리카ード内の隠し領域に記録されたMAC計算用の値の位置を特定する

値:1から0x190(400)

【0155】

MG (D) SERIAL-*nnn* (16バイト (upper: 8, Lower: 8))

意味: 記録機器のセキュリティブロック (セキュリティIC20) のシリアル番号

機能: 記録機器ごとに全て異なる固有の値

値: 0 から 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

【0156】

CONNUM (4バイト)

意味: コンテンツ累積番号

機能: 曲毎に累積されていく固有の値で記録機器のセキュリティブロックによって管理される。2の32乗、42億曲分用意されており、記録した曲の識別に使用する

値: 0 から 0xFFFFFFFF。

【0157】

YMDhms-S (4バイト) (Option)

意味: 再生制限付きのトラックの再生開始日時

機能: EMDで指定する再生開始を許可する日時

値: 上述した日時の表記と同じ

YMDhms-E (4バイト) (Option)

意味: 再生制限付きのトラックの再生終了日時

機能: EMDで指定する再生許可を終了する日時

値: 上述した日時の表記と同じ

【0158】

XCC (1バイト)

意味: 以下に説明するCCの拡張部

機能: コピー制御

【0159】

CT (1バイト) (Option)

意味：再生回数

機能：再生許可された回数の内で、実際に再生できる回数。再生の度にデクリメントする

値：0 x 0 0 ~ 0 x F F 未使用の時は、0 x 0 0 である

L T の b i t 7 が 1 で C T の値が 0 0 の場合は再生を禁止すること。

【 0 1 6 0 】

C C ( 1 バイト)

意味：COPY CONTROL

機能：コピー制御

値：図 2 5 に示すように、ビット 6 および 7 によってコピー制御情報を表し、ビット 4 および 5 によって高速デジタルコピーに関するコピー制御情報を表し、ビット 2 および 3 によってセキュリティブロック認証レベルを表す。ビット 0 および 1 は、未定義

C C の例：( b i t 7 , 6 ) 1 1 : 無制限のコピーを許可、0 1 : コピー禁止、0 0 : 1 回のコピーを許可 ( b i t 3 , 2 ) 0 0 : アナログないしデジタルインからの録音、M G 認証レベルは 0 とする

C D からのデジタル録音では ( b i t 7 , 6 ) は 0 0 、 ( b i t 3 , 2 ) は 0 0 となる

【 0 1 6 1 】

C N ( 1 バイト) ( O p t i o n )

意味：高速デジタルコピー H S C M S ( High speed Serial Copy Management System ) におけるコピー許可回数

機能：コピー 1 回か、コピーフリーかの区別を拡張し、回数で指定する。コピー第 1 世代の場合にのみ有効であり、コピーごとに減算する

値：0 0 : コピー禁止、0 1 から 0 x F E : 回数、0 x F F : 回数無制限。

【 0 1 6 2 】

上述したトラック情報領域 T R K I N F に続いて、0 x 0 3 7 0 から始まる 2 4 バイトのデータをパーツ管理用のパーツ情報領域 P R T I N F と呼び、1 つのトラックを複数のパーツで構成する場合に、時間軸の順番に P R T I N F を並べ

ていく。図 2 6 に P R T I N F の部分を示す。P R T I N F 内のデータについて、配置順序に従って以下に説明する。

【 0 1 6 3 】

P R T S I Z E ( 4 バイト)

意味：パーツサイズ

機能：パーツの大きさを表す。クラスタ：2 バイト（最上位）、開始 S U：1 バイト（上位）、終了 S U：1 バイト（最下位）

値：クラスタ：1 から 0 x 1 F 4 0 ( 8 0 0 0 )、開始 S U：0 から 0 x A 0 ( 1 6 0 )、終了 S U：0 から 0 x A 0 ( 1 6 0 )（但し、S U の数え方は、0 , 1 , 2 , と 0 から開始する）

【 0 1 6 4 】

P R T K E Y ( 8 バイト)

意味：パーツを暗号化するための値

機能：初期値 = 0、編集時は編集の規則に従うこと

値：0 から 0 x F F F F F F F F F F F F F F F F

【 0 1 6 5 】

C O N N U M 0 ( 4 バイト)

意味：最初に作られたコンテンツ累積番号キー

機能：コンテンツをユニークにするための I D の役割

値：コンテンツ累積番号初期値キーと同じ値とされる。

【 0 1 6 6 】

図 2 1 に戻る。A T R A C 3 データファイルの属性ヘッダ中には、図 2 1 に示すように、付加情報 I N F が含まれる。I N F は、トラックに関する付加情報データであり、ヘッダを伴った可変長の付加情報データ。複数の異なる付加情報が並べられることがある。それぞれに I D とデータサイズが付加されている。個々のヘッダを含む付加情報データは、最小 1 6 バイト以上で 4 バイトの整数倍の単位である。

【 0 1 6 7 】

上述した属性ヘッダに対して、A T R A C 3 データファイルの各ブロックのデ

ータが続く。図 2 7 に示すように、ブロック毎にヘッダが付加される。各ブロックのデータについて以下に説明する。

【 0 1 6 8 】

B L K I D - A 3 D ( 4 バイト)

意味 : BLOCKID FILE ID

機能 : A T R A C 3 データの先頭であることを識別するための値

値 : 固定値 = " A 3 D " ( 例えば 0 x 4 1 3 3 4 4 2 0 )

【 0 1 6 9 】

M C o d e ( 2 バイト)

意味 : MAKER CODE

機能 : 記録した機器の、メーカー、モデルを識別するコード

値 : 上位 1 0 ビット ( メーカーコード ) 下位 6 ビット ( 機種コード )

【 0 1 7 0 】

C O N N U M 0 ( 4 バイト)

意味 : 最初に作られたコンテンツ累積番号

機能 : コンテンツをユニークにするための I D の役割、編集されても値は変化させない

値 : コンテンツ累積番号初期値キーと同じ値とされる

【 0 1 7 1 】

B L O C K S E R I A L ( 4 バイト)

意味 : トラック毎に付けられた連続番号

機能 : ブロックの先頭は 0 から始まり次のブロックは + 1 ずつインクリメント  
編集されても値を変化させない

値 : 0 より始まり 0 x F F F F F F F F まで

【 0 1 7 2 】

B L O C K - S E E D ( 8 バイト)

意味 : 1 ブロックを暗号化するための 1 つの鍵

機能 : ブロックの先頭は、記録機器のセキュリティブロックで乱数を生成、続くブロックは、+ 1 インクリメントされた値、この値が失われると、1 ブロック

に相当する約1秒間、音が出せないために、ヘッダとブロック末尾に同じものが二重に書かれる。編集されても値を変化させない

値：初期は8バイトの乱数

【0173】

INITIALIZATION VECTOR (8バイト)

意味：ブロック毎にATRAC3データを暗号化、復号化する時に必要な初期値

機能：ブロックの先頭は0から始まり、次のブロックは最後のSUの最後の暗号化された8バイトの値。デバインドされたブロックの途中からの場合は開始SUの直前の最後の8バイトを用いる。編集されても値を変化させない

値：0から0xFFFFFFFFFFFFFFFF

【0174】

SU-nnn

意味：サウンドユニットのデータ

機能：1024サンプルから圧縮されたデータ、圧縮モードにより出力されるバイト数が異なる。編集されても値を変化させない（一例として、SPモードの時では、N=384バイト）

値：ATRAC3のデータ値。

【0175】

図21では、N=384であるので、1ブロックに42SUが書かれる。また、1ブロックの先頭の2つのスロット（4バイト）がヘッダとされ、最後の1スロット（2バイト）にBLKID-A3D、MCODE、CONNUM0、BLOCK SERIALが二重に書かれる。従って、1ブロックの余りの領域Mバイトは、 $(16, 384 - 384 \times 42 - 16 \times 3 = 208)$ （バイト）となる。この中に上述したように、8バイトのBLOCK SEEDが二重に記録される。

【0176】

ここで、フラッシュメモリ34に記憶されているデータは、後述するように例えば、ATRAC3方式で圧縮されている。圧縮の単位がサウンドユニットSU

である。従って、記憶装置 3 0 0 から再生装置 2 0 0 にデータを読み出す場合には、読み出しの最小単位は当該サウンドユニット S U となる。オーディオデータの圧縮方式は、A T R A C 3 などの A T R A C 方式以外の C O D E C 方式でもよい。

#### 【 0 1 7 7 】

ブロックシードデータ B S は、各ブロック毎に例えば乱数を発生して生成されたデータである。

#### 【 0 1 7 8 】

##### 〔フラッシュメモリ管理モジュール 3 5〕

フラッシュメモリ管理モジュール 3 5 は、フラッシュメモリ 3 4 へのデータの書き込み、フラッシュメモリ 3 4 からのデータの読み出しなどの制御を行う。

#### 【 0 1 7 9 】

図 1 5 に示す再生装置 2 0 0 の構成について説明する。再生装置 2 0 0 は、例えば、主制御モジュール 4 1、通信インターフェイス 4 2、制御モジュール 4 3、編集モジュール 4 4、圧縮／伸長モジュール 4 5、スピーカ 4 6、D／A変換器 4 7 および A／D変換器 4 8 を有する。

#### 【 0 1 8 0 】

##### 〔主制御モジュール 4 1〕

主制御モジュール 4 1 は、再生装置 2 0 0 の処理を統括的に制御する。

#### 【 0 1 8 1 】

##### 〔制御モジュール 4 3〕

図 1 5 に示すように、制御モジュール 4 3 は、例えば、乱数発生ユニット 6 0、記憶ユニット 6 1、鍵生成／鍵演算ユニット 6 2、相互認証ユニット 6 3、暗号化／復号ユニット 6 4 および制御ユニット 6 5 を有する。制御モジュール 4 3 は、制御モジュール 3 3 と同様に、シングルチップの暗号処理専用の集積回路であり、多層構造を有し、内部のメモリセルはアルミニウム層などのダミー層に挟まれている。また、制御モジュール 4 3 は、動作電圧または動作周波数の幅が狭く、外部から不正にデータを読み出せないように耐タンパー性を有している。乱数発生ユニット 6 0 は、乱数発生指示を受けると、6 4 ビット（8 バイト）の乱

数を発生する。記憶ユニット61は、認証処理に必要な種々のデータを記憶している。

#### 【0182】

鍵生成／鍵演算ユニット62は、例えば、ISO/IEC9797のMAC演算方式を用いた演算などの種々の演算を行って鍵データを生成する。このとき、"Block cipher Algorithm"としてFIPS PUB 46-2に規定されるDESが用いられる。

#### 【0183】

相互認証ユニット63は、例えば、コンピュータから入力したオーディオデータを記憶装置300に出力する動作を行うのに先立って、記憶装置300との間で相互認証処理を行う。また、相互認証ユニット63は、記憶装置300からオーディオデータを入力する動作を行うのに先立って、記憶装置300との間で相互認証処理を行う。また、相互認証ユニット63は、相互認証処理において、前述したMAC演算を行う。当該相互認証処理では、記憶ユニット61に記憶されているデータが用いられる。なお、相互認証ユニット63は、必要に応じて、例えば、パーソナルコンピュータ(PC)100あるいはネットワーク上のコンピュータとの間でオーディオデータの入出力を行う動作に先立って、パーソナルコンピュータ(PC)100あるいはネットワーク上のコンピュータとの間で相互認証処理を行う。

#### 【0184】

暗号化／復号ユニット64は、前述したように、FIPS PUB 81に規定されたECBモードおよびCBCモードを選択的に用いてブロック暗号化を行う。

#### 【0185】

暗号化／復号ユニット64は、FIPS 81のモードのうち、ECBモードおよびCBCモードの復号を選択的に行う。ここで、暗号化／復号ユニット64は、CBCモードにおいて、例えば56ビットの鍵データkを用いて、暗号文を、64ビットからなる暗号化ブロックを単位として復号して平文を生成する。

#### 【0186】

制御ユニット 6 5 は、乱数発生ユニット 6 0、記憶ユニット 6 1、鍵生成／鍵演算ユニット 6 2、相互認証ユニット 6 3 および暗号化／復号ユニット 6 4 の処理を統括的に制御する。

【 0 1 8 7 】

〔編集モジュール 4 4 〕

編集モジュール 4 4 は、例えば、図 1 6 に示すように記憶装置 3 0 0 のフラッシュメモリ 3 4 内に記憶されたトラックデータファイルを、ユーザからの操作指示に基づいて編集して新たなトラックデータファイルを生成する。

【 0 1 8 8 】

〔圧縮／伸長モジュール 4 5 〕

圧縮／伸長モジュール 4 5 は、例えば、記憶装置 3 0 0 から入力した暗号化されたオーディオデータを復号した後に再生する際に、A T R A C 3 方式で圧縮されているオーディオデータを伸長し、当該伸長したオーディオデータを D / A 変換器 4 7 に出力する。また、例えば、C D、D V D あるいは P C 1 から入力したオーディオデータを、記憶装置 3 0 0 に記憶する際に、当該オーディオデータを A T R A C 3 方式で圧縮する。

【 0 1 8 9 】

〔D / A 変換器 4 7 〕

D / A 変換器 4 7 は、圧縮／伸長モジュール 4 5 から入力したデジタル形式のオーディオデータをアナログ形式のオーディオデータに変換してスピーカ 4 6 に出力する。

【 0 1 9 0 】

〔スピーカ 4 6 〕

スピーカ 4 6 は、D / A 変換器 4 7 から入力したオーディオデータに応じた音響を出力する。

【 0 1 9 1 】

〔A / D 変換器 4 8 〕

A / D 変換器 4 8 は、例えば、C D プレーヤ 7 から入力したアナログ形式のオーディオデータをデジタル形式に変換して圧縮／伸長モジュール 4 5 に出力する

【0192】

〔メモリ49〕

メモリ49は、例えば、E2PROM（ex. フラッシュメモリ）であり、前述したキー有効化ブロック（EKB）、あるいはEKBに基づいて生成されるデバイスキーブロック（DKB）等の鍵データ、デバイス識別子としてのデバイスID等が格納される。

【0193】

〔コンテンツデータの記憶装置に対する格納処理および再生処理〕

図15に示す再生装置200と、記憶装置300の間では、コンテンツデータの移動、すなわち、再生装置200から記憶装置300のフラッシュメモリ34に対するデータ記録処理が実行され、さらに、記憶装置300のフラッシュメモリ34から再生装置200に対するデータ再生処理が実行される。

【0194】

このデータ記録および再生処理について、以下説明する。まず、再生装置200から記憶装置300のフラッシュメモリ34に対するデータ記録処理を図28のフローを用いて説明する。

【0195】

再生装置および記憶装置は、データ移動に先立ち、まずステップS2701、S2702に示す相互認証処理を実行する。図29に、共通鍵暗号方式を用いた相互認証方法（ISO/IEC 9798-2）を示す。図29においては、共通鍵暗号方式としてDESを用いているが、共通鍵暗号方式であれば他の方式も可能である。図29において、まず、Bが64ビットの乱数Rbを生成し、Rbおよび自己のIDであるID（b）をAに送信する。これを受信したAは、新たに64ビットの乱数Raを生成し、Ra、Rb、ID（b）の順に、DESのCBCモードで鍵Kabを用いてデータを暗号化し、Bに返送する。なお、鍵Kabは、AおよびBに共通の秘密鍵としてそれぞれの記録素子内に格納する鍵である。DESのCBCモードを用いた鍵Kabによる暗号化処理は、例えばDESを用いた処理においては、初期値とRaとを排他的論理和し、DES暗号化部において、鍵Ka

bを用いて暗号化し、暗号文E 1を生成し、続けて暗号文E 1とR bとを排他的論理和し、DES暗号化部において、鍵K a bを用いて暗号化し、暗号文E 2を生成し、さらに、暗号文E 2とID (b)とを排他的論理和し、DES暗号化部において、鍵K a bを用いて暗号化して生成した暗号文E 3とによって送信データ (Token-AB) を生成する。

## 【 0 1 9 6 】

これを受信したBは、受信データを、やはり共通の秘密鍵としてそれぞれの記録素子内に格納する鍵K a b (認証キー) で復号化する。受信データの復号化方法は、まず、暗号文E 1を認証キーK a bで復号化し、乱数R aを得る。次に、暗号文E 2を認証キーK a bで復号化し、その結果とE 1を排他的論理和し、R bを得る。最後に、暗号文E 3を認証キーK a bで復号化し、その結果とE 2を排他的論理和し、ID (b)を得る。こうして得られたR a、R b、ID (b)のうち、R bおよびID (b)が、Bが送信したものと一致するか検証する。この検証に通った場合、BはAを正当なものとして認証する。

## 【 0 1 9 7 】

次にBは、認証後に使用するセッションキー (K s e s) を生成する (生成方法は、乱数を用いる)。そして、R b、R a、K s e sの順に、DESのCBCモードで認証キーK a bを用いて暗号化し、Aに返送する。

## 【 0 1 9 8 】

これを受信したAは、受信データを認証キーK a bで復号化する。受信データの復号化方法は、Bの復号化処理と同様であるので、ここでは詳細を省略する。こうして得られたR b、R a、K s e sの内、R bおよびR aが、Aが送信したものと一致するか検証する。この検証に通った場合、AはBを正当なものとして認証する。互いに相手を認証した後には、セッションキーK s e sは、認証後の秘密通信のための共通鍵として利用される。

## 【 0 1 9 9 】

なお、受信データの検証の際に、不正、不一致が見つかった場合には、相互認証が失敗したものとして処理を終了 (S 2 7 0 3でN o) する。

## 【 0 2 0 0 】

相互認証が成立（S 2 7 0 3 で Y e s）した場合は、ステップ S 2 7 0 4 において、再生装置がコンテンツキー K c o n の生成処理を実行する。この処理は、図 1 5 の乱数生成ユニット 6 0 で生成した乱数を用いて鍵生成／鍵演算ユニット 6 2 において実行される。

#### 【 0 2 0 1 】

次に、ステップ S 2 7 0 5 において、（１）コンテンツキー K c o n を有効化キーブロック（E K B）から取得される暗号化キー K E K を用いて暗号化処理して、E（K E K，K c o n）を生成するとともに、（２）コンテンツキー K c o n を認証処理において生成したセッションキー（K s e s）で暗号化処理を実行して、E（K s e s，K c o n）を生成して、記憶装置（メモリカード）に送信する。

#### 【 0 2 0 2 】

ステップ S 2 7 0 6 では、記憶装置が再生装置から受信した E（K s e s，K c o n）をセッションキーで復号してコンテンツキー K c o n を取得し、さらに、K c o n を記憶装置に予め格納されているストレージキー K s t m によって暗号化して E（K s t m，K c o n）を生成し、これを再生装置に送信する。

#### 【 0 2 0 3 】

次に、再生装置は、ステップ S 2 7 0 7 において、ステップ S 2 7 0 5 で生成した E（K E K，K c o n）、およびステップ S 2 7 0 6 で記憶装置から受信した E（K s t m，K c o n）を用いて、データファイル（図 2 1 参照）を構成するトラック情報領域 T R K I N F データを生成し、データファイルのフォーマット処理の後、これを記憶装置（メモリカード）に送信する。

#### 【 0 2 0 4 】

ステップ S 2 7 0 8 において、記憶装置（メモリカード）は、再生装置から受信したデータファイルをフラッシュメモリに格納する。

#### 【 0 2 0 5 】

このような処理により、データファイルのトラック情報領域 T R K I N F データには、先に説明した図 2 1、図 2 3 に示すように、コンテンツキー K c o n を有効化キーブロック（E K B）から取得される暗号化キー K E K を用いて暗号化

処理した E ( K E K , K c o n ) と、コンテンツキー K c o n を記憶装置に予め格納されているストレージキー K s t m によって暗号化した E ( K s t m , K c o n ) の 2 つの暗号化コンテンツキーが格納されることになる。

## 【 0 2 0 6 】

なお、音楽データ、画像データ等の暗号化処理は、コンテンツキー K c o n をそのままコンテンツの暗号化鍵として適用して実行するか、あるいはコンテンツを構成するパーツ、またはブロック等を単位として、コンテンツキーと他のキー生成データに基づいて各パーツ単位、またはブロック単位の暗号化鍵を個別に生成して各パーツ単位、またはブロック単位の暗号化処理を行なう構成とすることが可能である。

## 【 0 2 0 7 】

このようなデータファイルを用いた再生処理においては、再生装置は、 E ( K E K , K c o n ) と、 E ( K s t m , K c o n ) のいずれかを選択的に適用してコンテンツキー K c o n を取得可能となる。

## 【 0 2 0 8 】

次に、再生装置 2 0 0 が記憶装置 3 0 0 のフラッシュメモリ 3 4 に格納されたデータの読み出し処理、すなわち再生処理を実行する場合の処理を図 3 0 のフローを用いて説明する。

## 【 0 2 0 9 】

再生装置および記憶装置は、データ移動に先立ち、まずステップ S 2 9 0 1、S 2 9 0 2 に示す相互認証処理を実行する。この処理は、先に説明した図 2 9 の処理と同様である。相互認証が失敗した場合 ( S 2 9 0 3 で N o ) は、処理を終了する。

## 【 0 2 1 0 】

相互認証が成立 ( S 2 9 0 3 で Y e s ) した場合は、ステップ S 2 9 0 4 において、記憶装置が再生装置に対してデータファイルを送信する。データファイルを受信した再生装置は、データファイル中のトラック情報領域 T R K I N F データを検査し、コンテンツキー ( K c o n ) の格納状況を判別する。この判別処理は、キー有効化ブロック ( E K B ) によって取得される暗号化キー K E K によっ

て暗号化されたコンテンツキー、すなわち  $E(K_{EK}, K_{con})$  が格納されているか否かを判別する処理である。 $E(K_{EK}, K_{con})$  の有無は、先の図 21, 23 で説明したデータファイル中のトラック情報領域  $TRKINF$  データの  $[EKI]$  のデータにより判別可能である。

## 【0211】

$E(K_{EK}, K_{con})$  が格納されている場合（ステップ S2906 で Yes）は、ステップ S2907 に進み、キー有効化ブロック（ $EKB$ ）の処理により、暗号化キー  $K_{EK}$  を取得して、取得した暗号化キー  $K_{EK}$  により、 $E(K_{EK}, K_{con})$  を復号して、コンテンツキー  $K_{con}$  を取得する。

## 【0212】

$E(K_{EK}, K_{con})$  が格納されていない場合（ステップ S2906 で No）は、ステップ S2908 において、記憶装置の制御モジュール 33 において、記憶装置に予め格納されているストレージキー  $K_{stm}$  によって暗号化した  $E(K_{stm}, K_{con})$  をストレージキー  $K_{stm}$  によって復号して、さらに、相互認証処理において再生装置および記憶装置で共有したセッションキー  $K_{ses}$  で暗号化したデータ  $E(K_{ses}, K_{con})$  を生成して、再生装置に送信する。

## 【0213】

再生装置は、ステップ S2909 において、記憶装置から受信した  $E(K_{ses}, K_{con})$  をセッションキー  $K_{ses}$  で復号してコンテンツキー  $K_{con}$  を取得する。

## 【0214】

ステップ S2910 では、ステップ S2907、またはステップ S2909 のいずれかにおいて取得したコンテンツキー  $K_{con}$  により暗号化コンテンツの復号を行なう。

## 【0215】

このように、暗号化コンテンツの再生処理において、再生装置は、 $E(K_{EK}, K_{con})$  を有効化キーブロック（ $EKB$ ）から取得される暗号化キー  $K_{EK}$  を用いて復号するか、または、記憶装置に予め格納されているストレージキー  $K$

s t mによって暗号化したE (K s t m, K c o n)に基づく処理を実行するか、いずれかの処理を実行することによりコンテンツキーK c o nを取得することができる。

## 【 0 2 1 6 】

なお、音楽データ、画像データ等の復号処理は、コンテンツキーK c o nをそのままコンテンツの復号鍵として適用して実行するか、あるいはコンテンツを構成するパーツ、またはブロック等を単位として、コンテンツキーと他のキー生成データに基づいて各パーツ単位、またはブロック単位の復号鍵を個別に生成して各パーツ単位、またはブロック単位の復号処理を行なう構成とすることが可能である。

## 【 0 2 1 7 】

## [ K E K を格納した E K B のフォーマット ]

先に図6を用いて有効化キーブロック (E K B) の概略的なフォーマットについて説明したが、さらに、キー暗号化キー (K E K) を有効化キーブロック (E K B) に格納して保持する場合の具体的なデータ構成例について説明する。

## 【 0 2 1 8 】

図31にキー暗号化キー (K E K) を有効化キーブロック (E K B) に格納したデータであるE K Bである配信鍵許可情報ファイルの構成例を示す。デバイス (再生装置) は、このファイルから必要に応じてキー暗号化キー (K E K) を取り出して、K E KによりE (K E K, K c o n) を復号してコンテンツキー: K c o nを取得してコンテンツの復号を実行する。各データについて説明する。

## 【 0 2 1 9 】

## B L K I D - E K B ( 4 バイト )

意味: BLOCKID FILE ID

機能: 配信鍵情報ファイルの先頭であることを識別するための値

値: 固定値 = " E K B " (例えば 0 x 4 5 4 B 4 2 2 0)

## 【 0 2 2 0 】

## M C o d e ( 2 バイト )

意味: MAKER CODE

機能：記録した機器の、メーカー、モデルを識別するコード

値：上位10ビット（メーカーコード） 下位6ビット（機種コード）

【0221】

L K F

意味：LINK FILE INFORMATION

機能：このE K Bによって取得されるK E Kが適用可能なコンテンツデータであるリンクファイルを識別する。

値：0～0xFF

bit 7：再生管理ファイル（P B L I S T）に使用：1、未使用：0

bit 6：改竄チェック値（I C V）に使用：1、未使用：0

bit 5～0：リザーブ

【0222】

L I N K   c o u n t

意味：LINK COUNT

機能：リンクしているファイル（例えばA T R A C K 3ファイル）数

値：0～0xFFFFFFFF

【0223】

V e r s i o n

意味：VERSION

機能：配信鍵許可情報ファイルのバージョンを示す。

値：0～0xFFFFFFFF

【0224】

E A

意味：Encryption Algorithm

機能：配信鍵許可情報ファイルのトレース処理アルゴリズムを示す。

値：0～0xFF

00h：3DES：トリプルDESモードによる処理

01h：DES：シングルDESモードによる処理

なお、トリプルDESモードによる処理は、2種類以上の暗号処理キーを用い

る暗号処理であり、シングルDESモードは1つのキーによる処理である。

【0225】

KEK1

意味: Key Encrypting Key

機能: キー有効化ブロック (EKB) 中のルートキー (最上位) キーで暗号化されたコンテンツキー暗号キー

値: 0~0xFFFFFFFFFFFFFFFF

【0226】

KEK2

意味: Key Encrypting Key

機能: キー有効化ブロック (EKB) 中のルートキー (最上位) キーで暗号化されたコンテンツキー暗号キー

値: 0~0xFFFFFFFFFFFFFFFF

【0227】

E (Version)

意味: Encrypted Version

機能: キー有効化ブロック (EKB) 中のルートキー (最上位) キーで暗号化されたバージョン番号。復号時の下4バイトはリザーブ

値: 0~0xFFFFFFFFFFFFFFFF

【0228】

Size of tag part

意味: Size of tag part

機能: 配信鍵許可情報ファイルを構成するデータのタグ部分のサイズ (Byte)

値: 0~0xFFFFFFFF

【0229】

Size of Key part

意味: Size of key part

機能: 配信鍵許可情報ファイルを構成するデータのキー部分のサイズ (Byte)

e)

値：0～0xFFFFFFFF

【0230】

Size of Sign part

意味：Size of sign part

機能：配信鍵許可情報ファイルを構成するデータのサイン部分のサイズ (Byte)

値：0～0xFFFFFFFF

【0231】

Tag part

意味：Tag part

機能：配信鍵許可情報ファイルを構成するデータのタグ部分のデータ

値：すべての値

8バイトに満たない場合は0で埋めて8バイトにする。

【0232】

Key part

意味：Key part

機能：配信鍵許可情報ファイルを構成するデータのキー部分のデータ

値：すべての値

【0233】

Signature part

意味：Signature part

機能：配信鍵許可情報ファイルを構成するデータの署名 (Signature) 部分のデータ

値：すべての値

【0234】

上述の説明および図31によって示されるように、デバイスに対して提供される配信鍵許可情報ファイルには、その配信鍵許可情報ファイルから取得されるKEKが適用可能なコンテンツデータであるリンクファイルを識別するための識別

データ [LKF] が格納され、さらに、リンクしているファイル（例えば ATRACK3 ファイル）数としてのデータ [Link Count] が格納される。再生装置は、[LKF]、[Link Count] を参照することにより、その配信鍵許可情報ファイルから取得される KEK を適用すべきデータが存在するか否かおよびその数を知ることが可能となる。

## 【0235】

[リンク情報を用いたデータ復号、再生処理]

上述した配信鍵許可情報ファイルに含まれるリンクファイルを識別するための識別データ [LKF]、リンクしているファイル（例えば ATRACK3 ファイル）数としてのデータ [Link Count] を用いて、効率的にデータの復号、再生を実行する処理態様について、以下説明する。

## 【0236】

図32に記憶装置のデータ格納領域、例えば図15に示す記憶装置300のフラッシュメモリ34に格納されたデータファイル構成例を示す。ここでは、音楽データ（HIFI）のディレクトリ構成のみを例として示しているが、さらに、画像ファイル等のディレクトリが存在してもよい。

## 【0237】

図32に示す音楽データのディレクトリには、再生管理ファイル（PBLIST）、暗号化コンテンツとして複数のATRACK3データファイル（A3D）が含まれる。さらに、記憶装置には、複数の有効化キープロックファイル（EKB<sub>n</sub>）が格納される。ATRACK3データファイル（A3D）の復号処理に適用するコンテンツキーを取得するための有効化キープロックファイル（EKB<sub>n</sub>）は、ATRACK3データファイル（A3D）に含まれるポインタによって判別される。図32に示すように、1つの有効化キープロックファイル（EKB1）3101は複数（3）のATRACK3データファイル（A3D）の復号処理に適用される。

## 【0238】

この場合、有効化キープロックファイル（EKB1）3101に対応する配信鍵許可情報ファイルの [Link Count] には3つのコンテンツに適用さ

れることを示すデータが格納されることになる。

#### 【 0 2 3 9 】

図 3 2 のような複数のコンテンツファイル、複数の有効化キーブロックファイルを格納した記憶装置であるメモリカードからコンテンツを復号して、再生する場合の処理フローを図 3 3 に示す。

#### 【 0 2 4 0 】

図 3 3 の処理は、例えば記憶装置としてのメモリカードを再生装置にセットした際、あるいはメモリカードを装着した再生装置の電源を ON した際に再生装置が実行する処理である。

#### 【 0 2 4 1 】

まず、ステップ S 3 2 0 1 において、再生装置は、各々の E K B ファイルのトラック情報を読み取り、[ L i n c   C o u n t ] をチェックする。さらに、[ L i n c   C o u n t ] のカウント数が多いものから順に、予め定められた個数 [ n ] の E K B ファイルを選択する。個数 [ n ] は、再生装置の所定メモリ領域、すなわちキー暗号化キー：K E K を格納保持する領域に格納可能な個数に相当する個数として設定される。

#### 【 0 2 4 2 】

次に、ステップ S 3 2 0 2 において、選択した E K B の処理により、複数 [ n ] のキー暗号化キー：K E K を取得し、これらを再生装置の鍵格納領域として設定された R A M の所定領域に格納する。

#### 【 0 2 4 3 】

次に、再生装置は、ステップ S 3 2 0 3 において、復号、再生するコンテンツを選択する。さらに、ステップ S 3 2 0 4 において、その選択コンテンツの復号に適用する K E K が R A M に格納されているか否かを判定し、Y e s の場合は、ステップ S 3 2 0 5 に進み、その対応 K E K に基づいて、E ( K E K , K c o n ) を復号してコンテンツキーを取得して、ステップ S 3 2 0 9 で再生、すなわち、取得したコンテンツキーによるデータの復号、再生処理を実行する。

#### 【 0 2 4 4 】

ステップ S 3 2 0 4 において、選択コンテンツの復号に適用する K E K が R A

Mに格納されていない場合は、ステップS3206において、ストレージキーで暗号化されたコンテンツキー、すなわち、 $E(K_{stm}, K_{con})$ の有無を判定し、ある場合は、ステップS3207において、 $E(K_{stm}, K_{con})$ の復号処理によりコンテンツキーを取得して、ステップS3209で再生、すなわち、取得したコンテンツキーによるデータの復号、再生処理を実行する。

## 【0245】

また、ステップS3206において、 $E(K_{stm}, K_{con})$ がないと判定されると、その復号対象コンテンツに適用すべきEKBを記憶装置から取得して、取得したEKBの復号処理によりKEKを取得し、取得したKEKによる $E(KEK, K_{con})$ の復号処理を実行してコンテンツキーを取得して、ステップS3209で再生、すなわち、取得したコンテンツキーによるデータの復号、再生処理を実行する。

## 【0246】

このように、再生装置は、予め記憶装置に格納した複数のキー有効化ブロック(EKB)の[Line Count]をチェックし、[Line Count]のカウント数が多いEKBの復号を実行して、キー暗号化キー:KEKを格納しておく構成とすることにより、コンテンツ再生処理の際に、高い確率でRAMに格納したKEKを適用可能となり、効率的なコンテンツ再生が実行できる。

## 【0247】

## [キー有効化ブロック(EKB)による認証キー配信]

上述の有効化キーブロック(EKB)を使用したキーの配信において、認証処理を実行する際に使用する認証キーIKnを配信することにより、安全な秘密鍵として共有する認証キーを提供し、共通鍵方式に従った認証処理を実行する構成について説明する。

## 【0248】

共通鍵暗号方式を用いた相互認証方法(ISO/IEC 9798-2)は、先に図29を用いて説明した処理であり、データ送受信が実行される前の処理として、双方の正当性を確認するための処理として実行される。認証処理においては、データの送受信を行なう、例えば再生装置と記憶装置は認証キーKabを共有する。この共

通鍵  $K_{ab}$  を上述の有効化キープブロック (EKB) を使用して再生装置に配信する。

#### 【0249】

図34および図35に複数のデバイスに共通の認証キー  $IK_n$  を有効化キープブロック (EKB) によって配信する構成例を示す。図34はデバイス0, 1, 2, 3に対して復号可能な認証キー  $IK_n$  を配信する例、図35はデバイス0, 1, 2, 3中のデバイス3をリボーク (排除) してデバイス0, 1, 2に対してのみ復号可能な認証キーを配信する例を示す。

#### 【0250】

図34の例では、更新ノードキー  $K(t)_{00}$  によって、認証キー  $IK_n$  を暗号化したデータ (b) とともに、デバイス0, 1, 2, 3においてそれぞれの有するノードキー、リーフキーを用いて更新されたノードキー  $K(t)_{00}$  を復号可能な有効化キープブロック (EKB) を生成して配信する。それぞれのデバイスは、図34の右側に示すようにまず、EKBを処理 (復号) することにより、更新されたノードキー  $K(t)_{00}$  を取得し、次に、取得したノードキー  $K(t)_{00}$  を用いて暗号化された認証キー:  $Enc(K(t)_{00}, IK_n)$  を復号して認証キー  $IK_n$  を得ることが可能となる。

#### 【0251】

その他のデバイス4, 5, 6, 7…は同一の有効化キープブロック (EKB) を受信しても自身の保有するノードキー、リーフキーでは、EKBを処理して更新されたノードキー  $K(t)_{00}$  を取得することができないので、安全に正当なデバイスに対してのみ認証キーを送付することができる。

#### 【0252】

一方、図35の例は、デバイス3が、例えば鍵の漏洩によりリボーク (排除) されているとして、他のグループのメンバ、すなわち、デバイス0, 1, 2, に対してのみ復号可能な有効化キープブロック (EKB) を生成して配信した例である。図35に示す (a) 有効化キープブロック (EKB) と、(b) 認証キー ( $IK_n$ ) をノードキー ( $K(t)_{00}$ ) で暗号化したデータを配信する。

#### 【0253】

図 3 5 の右側には、復号手順を示してある。デバイス 0, 1, 2 は、まず、受領した有効化キープブロックから自身の保有するリーフキーまたはノードキーを用いた復号処理により、更新ノードキー ( $K(t)00$ ) を取得する。次に、 $K(t)00$  による復号により認証キー  $IK_n$  を取得する。

## 【 0 2 5 4 】

他のグループのデバイス、例えばデバイス 4, 5, 6 … は、この同様のデータ ( $EKB$ ) を受信したとしても、自身の保有するリーフキー、ノードキーを用いて更新ノードキー ( $K(t)00$ ) を取得することができない。同様にリボークされたデバイス 3 においても、自身の保有するリーフキー、ノードキーでは、更新ノードキー ( $K(t)00$ ) を取得することができず、正当な権利を有するデバイスのみが認証キーを復号して利用することが可能となる。

## 【 0 2 5 5 】

このように、 $EKB$  を利用した認証キーの配送を用いれば、データ量を少なくして、かつ安全に正当権利者のみが復号可能とした認証キーを配信することが可能となる。また、有効化キープブロック ( $EKB$ ) によって暗号化され提供される  $EKB$  配信認証鍵は、世代 (バージョン) 管理がなされ、世代毎の更新処理が実行され、任意のタイミングでのデバイスのリボーク (排除) が可能である。

## 【 0 2 5 6 】

上述した  $EKB$  による認証キーの提供処理により、リボークされたデバイス (再生装置) では、記憶装置 (例えばメモリカード) との認証処理が成立せず、データの不正な復号が不可能となる。

## 【 0 2 5 7 】

さらに、 $EKB$  を利用した認証キーの配送を用いれば、メモリカード以外の記憶媒体、例えば再生装置に内蔵したハードディスク等の記憶媒体に対するデータ格納、再生処理に対する制御も可能となる。

## 【 0 2 5 8 】

先の図 2 8 ~ 3 0 を用いて説明したように、記憶装置を利用したコンテンツの記録、再生処理においては、相互認証処理が実行され、相互認証処理の成立を条件として、データの記録および再生が可能となる。この認証処理プログラムは、

メモリカードのような相互認証処理が可能な記憶装置との間での処理においては有効に作用するが、例えば、再生装置がハードディスク、CD-R等、暗号処理機能を持たない、すなわち相互認証を実行不可能な記憶媒体に対してデータ格納、データ再生時には意味をなさないことになる。しかし、本発明のシステムでは、このような認証不可能な機器を利用したデータ格納、あるいはデータ再生処理においても認証処理プログラムを実行させる構成とする。ハードディスク、CD-R等は相互認証が不可能であるので、仮想のメモリカード（メモリスティック）を再生装置に構成し、仮想メモリカードと再生装置間において認証処理を実行させて、認証成立を条件として、認証機能を持たない記憶媒体に対するデータ格納処理、あるいは記憶媒体からのデータ再生を可能とする。

## 【0259】

これらの仮想メモリカードを使用したデータ記録、再生処理フローを図36に示す。まず、再生装置は、再生装置内の仮想メモリカードとの間で相互認証処理を実行する。ステップS3502において、認証成立したか否かを判定し、成立したことを条件としてステップS3503に進み、認証機能を持たない記憶媒体、例えばハードディスク、CD-R、DVD等を用いたデータ記録、再生処理を実行する。

## 【0260】

ステップS3502において、認証が成立しなかったと判定された場合は、ステップS3503の認証機能を持たない記憶媒体、例えばハードディスク、CD-R、DVD等を用いたデータ記録、再生処理が実行されない。

## 【0261】

ここで、仮想メモリカードには、予め、先の図16で説明した認証鍵データを格納した構成とし、再生装置が使用する認証キーを前述したように、キー有効化ブロックで提供する構成とする。

## 【0262】

このように、再生装置の認証キーをキー有効化ブロック（EKB）で提供することにより、正当なライセンスを持つデバイス（再生装置）に対してのみ、仮想メモリカードとの相互認証可能な認証キーを配信することが可能となる。従って

、不正な機器、すなわちリボークされた再生装置には、有効な認証キーが配信しない処理が可能となる。有効な認証キーが提供されない再生装置は、相互認証が不成立となり、認証機能を持つメモリカードのみならず、認証機能を持たない記憶媒体、例えばハードディスク、CD-R、DVD等を用いたデータ記録、再生処理が実行されず、不正な機器によるデータ記録、再生を排除することが可能となる。

## 【0263】

すなわち、認証鍵を提供する有効化キープロック（EKB）をキーツリーのリーフを構成するデータ処理装置中、正当なライセンスを持つデータ処理装置においてのみ復号可能で、正当ライセンスを持たない不正なデータ処理装置においては復号不可能な有効化キープロック（EKB）として提供することにより、不正なデータ処理装置における仮想メモリデバイスとの認証成立を防止して、不正データ処理装置におけるコンテンツ利用を排除可能とした構成を有するライセンスシステムが実現される。

## 【0264】

【チェック値（ICV: Integrity Check Value）格納構成】

次に、コンテンツの改竄を防止するためにコンテンツのインテグリティ・チェック値（ICV）を生成して、コンテンツに対応付けて、ICVの計算により、コンテンツ改竄の有無を判定する処理構成について説明する。

## 【0265】

コンテンツのインテグリティ・チェック値（ICV）は、例えばコンテンツに対するハッシュ関数を用いて計算され、 $ICV = hash(K_{icv}, C1, C2, \dots)$  によって計算される。 $K_{icv}$  は ICV 生成キーである。 $C1$ 、 $C2$  はコンテンツの情報であり、コンテンツの重要情報のメッセージ認証符号（MAC: Message authentication Code）が使用される。前述したように、[MAC] は、図 21 で説明した ATRAC3 データファイルにも含まれる。これらを使用してインテグリティ・チェック値（ICV）の計算がなされる。

## 【0266】

DES 暗号処理構成を用いた MAC 値生成例を図 37 に示す。図 37 の構成に

示すように対象となるメッセージを8バイト単位に分割し、(以下、分割されたメッセージをM1、M2、・・・、MNとする)、まず、初期値 (Initial Value (以下、IVとする)) とM1を排他的論理和する(その結果をI1とする)。次に、I1をDES暗号化部に入れ、鍵(以下、K1とする)を用いて暗号化する(出力をE1とする)。続けて、E1およびM2を排他的論理和し、その出力I2をDES暗号化部へ入れ、鍵K1を用いて暗号化する(出力E2)。以下、これを繰り返し、全てのメッセージに対して暗号化処理を施す。最後に出てきたENがメッセージ認証符号(MAC (Message Authentication Code))となる。なお、メッセージとしては、検証対象となるコンテンツおよびヘッダ情報等のコンテンツ関連データを構成する部分データが使用可能である。

## 【0267】

このようなコンテンツのMAC値とICV生成キーK<sub>icv</sub>にハッシュ関数を適用して用いてコンテンツのインテグリティ・チェック値(ICV)が生成される。改竄のないことが保証された例えばコンテンツ生成時に生成したICVと、新たにコンテンツに基づいて生成したICVとを比較して同一のICVが得られればコンテンツに改竄のないことが保証され、ICVが異なれば、改竄があったと判定される。

## 【0268】

上述のようなインテグリティ・チェック値(ICV)は、コンテンツ個々に対して生成される複数のコンテンツMAC値により、1つのインテグリティ・チェック値(ICV)を生成することが可能である。複数のMACによるICVの計算は、例えば、 $ICV = MAC(K_{icv}, C\_MAC[0] || C\_MAC[1] || C\_MAC[2] || \dots)$  によって生成する。

## 【0269】

コンテンツ生成時に生成したICVを格納しておき、チェック処理時に生成ICVと格納ICVの比較処理を行なう。両ICVが一致すれば改竄無しと判定し、ICVが不一致の場合は、改竄が有りと判定され、データ再生等の処理制限がなされる。

## 【0270】

メモ리카ード等の記憶装置には、音楽コンテンツのみならず、画像データ、ゲームプログラムデータ等、カテゴリの異なるが格納される。これら各カテゴリのコンテンツも改竄の防止を図るため、各カテゴリ毎にインテグリティ・チェック値（ICV）を生成して格納することがコンテンツ改竄チェックのためには有効な手段となる。

## 【0271】

しかしながら、メモリに格納するコンテンツ数が増大すると、検証用のチェック値を正規のコンテンツデータに基づいて生成し、格納し管理することが困難となる。特に、昨今フラッシュメモリを使用したメモ리카ード等の容量の大きい媒体においては、音楽データ、画像データ、プログラムデータ等、様々なカテゴリのコンテンツデータがメモリに格納されることとなる。このような環境においては、チェック値の生成処理、格納処理、改竄チェック処理の管理は困難となる。格納データ全体に対するチェック値を生成すると、チェック対象となったデータ全体に対するチェック値生成処理を実行することが必要となる。例えばDES-CBCモードにおいて生成されるメッセージ認証符号（MAC）により、チェック値ICVを求める手法を行なう場合、データ全体に対するDES-CBCの処理を実行することが必要となる。この計算量は、データ長が長くなるにつれ増大することとなり、処理効率の点で問題がある。

## 【0272】

記憶装置として使用可能なメモ리카ードには、多くのカテゴリの異なるコンテンツが格納される。これらのカテゴリの異なるコンテンツの改竄チェック管理をカテゴリ毎に独立したインテグリティ・チェック値（ICV）を生成して実行する構成とすることにより、ICVのチェック時、あるいはICVの変更時、例えばデータ変更時の新たなインテグリティ・チェック値（ICV）の生成処理が、1つのカテゴリ内のデータを対象として実行可能となり、他のカテゴリに影響を及ぼすことがない。このようにカテゴリ毎の複数のインテグリティ・チェック値（ICV）を格納する構成について説明する。

## 【0273】

図38に記憶装置に格納されるデータ構成と、それぞれのインテグリティ・チ

チェック値（ICV）の格納構成例を示す。メモリカード等の記憶部（フラッシュメモリ）には、図38に示されるように音楽データのディレクトリに、再生管理ファイル（PBLIST）、暗号化コンテンツとして複数のATRACK3データファイル（A3D）が含まれ、さらに、メモリには、複数のカテゴリに属するコンテンツデータ（#1～#n）が格納される。複数のカテゴリとは、例えば、音楽データ、画像データ、ゲームプログラム等である。さらに、同様の画像データであっても、それぞれのデータ提供元に応じて別のディレクトリとして独立のカテゴリとして管理してもよい。

## 【0274】

また、前述の有効化キープブロック（EKB）の管理単位（エンティティ）を1カテゴリとして設定してもよい。すなわち、ある有効化キープブロック（EKB）によって取得されるキー暗号キー：KEKによって復号されるコンテンツキーKconを適用可能なコンテンツ集合を1つのカテゴリとして設定してもよい。

## 【0275】

再生管理ファイル（PBLIST）、暗号化コンテンツとして複数のATRACK3データファイル（A3D）の各々には、改竄チェックのためのメッセージ認証符号（MAC（Message Authentication Code））が含まれ、これらのMAC値に基づいてインテグリティ・チェック値（ICV（con））が生成される。複数のコンテンツのMAC値は、フラッシュメモリのシーケンスページにMACリストとして格納、管理され、これらのMACリストに基づいてICV生成キーKicvを適用して得られるインテグリティ・チェック値（ICV（con））が格納保存される。

## 【0276】

コンテンツMAC値を格納するシーケンスページフォーマットを図39に示す。シーケンスページ領域は、一般コンテンツデータの書き込み禁止領域として設定された領域である。図39のシーケンスページ構成について説明する。

## 【0277】

E（kSTR，kCON）は、メモリカードのストレージキーで暗号化したコンテンツキーである。ID（upper），（lower）は、メモリカードの

識別子 (ID) の格納領域である。C\_MAC [0] は、再生管理ファイル (PBLIST) の構成データに基づいて生成された MAC 値である。C\_MAC [1] は、コンテンツ、例えば ATRACK3 データファイル #1 のデータに基づいて生成された MAC 値、以下、コンテンツ毎に MAC 値が格納される。これらの MAC 値に基づいてインテグリティ・チェック値 (ICV (con)) が生成され、生成された ICV (con) がシリアルプロトコルを通してメモリに書き込まれる。なお、異なる鍵システムに対応するため、それぞれの鍵システムから生成される ICV をそれぞれ違うエリアに格納する構成とすることが好ましい。

## 【0278】

また、カテゴリ毎に改竄チェックのために生成される各カテゴリ毎のインテグリティ・チェック値 (ICV) は、メモ리카ードの記憶部 (フラッシュメモリ) のプールページに記録される。プールページもまた、一般データの書き込みの禁止された領域として設定されている。

## 【0279】

各カテゴリ毎のインテグリティ・チェック値 (ICV) を格納するプールページフォーマットを図40に示す。#0\_revision は、カテゴリ #0 の更新データが設定され、更新された場合はインクリメントされる。#0\_version は、カテゴリ #0 のバージョン、#0\_E (KEK, Kicv) は、カテゴリ #0 のキー暗号化キー (KEK) で暗号化した ICV 生成キー (Kicv) であり、ICV0 は、カテゴリ #0 のインテグリティ・チェック値 (ICV) 値である。以下、同様のデータが各カテゴリ毎に EKB #15 まで格納可能となっている。

## 【0280】

ICV のチェックは、パワーオン時、またはメモ리카ード等の記憶装置が再生装置にセットされたことを条件として開始される。図41に ICV チェックを含む処理フローを示す。

## 【0281】

まず、再生装置がパワーオン、または新たなメモ리카ード等が装着されたことを検知すると、ステップ S4001 において、再生装置と記憶装置間の相互認証

が可能か否かが判定され、可能である場合は、ステップ S 4 0 0 2 において記憶装置と再生装置間での相互認証処理（図 2 9 参照）が実行される。また、ステップ S 4 0 0 1 において、再生装置と記憶装置間の相互認証が可能でないと判定された場合は、ステップ S 4 0 0 3 において、前述した仮想メモリカードと再生装置間の相互認証処理が実行される。

## 【 0 2 8 2 】

ステップ S 4 0 0 4 で相互認証が成立したか否かが判定され、不成立の場合は、以下の処理は実行されないで終了する。相互認証が成立の場合は、ステップ S 4 0 0 5 において I C V の計算が実行される。I C V は、前述したように各ファイルの M A C 値に基づいて算出される。

## 【 0 2 8 3 】

次にステップ S 4 0 0 6 において、計算によって算出された生成 I C V と、予め格納してある格納 I C V との比較が実行される。両 I C V が一致した場合は、データ改竄がないと判定され、ステップ S 4 0 0 7 において、データ再生等の様々な処理が実行される。一方、I C V が不一致であった場合は、データ改竄があると判定され、データの再生等を行わず処理を終了する。このような処理を実行することによりデータ改竄の防止、改竄されたデータの再生が排除される。

## 【 0 2 8 4 】

このように、カテゴリの異なるコンテンツについて、カテゴリ毎に独立したインテグリティ・チェック値（I C V）を生成して管理する構成とすることにより、I C V のチェック時、あるいは I C V の変更時、例えばデータ変更時の新たなインテグリティ・チェック値（I C V）の生成処理が、1 つのカテゴリ内のデータを対象として実行可能となり、他のカテゴリに影響を及ぼすことがない。

## 【 0 2 8 5 】

## 〔拡張 M A C 構成〕

前述の再生管理ファイルまたは、A T R A C K 3 データファイルのデータ内容の欄で説明したデータ改竄チェック用の M A C (Message Authentication Code) の生成、および各ファイルに対する格納処理の変形例として、拡張 M A C の生成、格納処理について、以下説明する。

## 【 0 2 8 6 】

図 4 2 に拡張 MAC の生成、格納処理例を示す。図 4 2 には、先の図 2 1 ～ 2 3 で示した A T R A C K 3 データファイルの一部を示している。データ改竄チェック用の MAC (Message Authentication Code) は、例えば A T R A C K 3 データファイル中のいくつかのデータ項目のデータに基づいて、先の図 3 7 で説明した処理によって生成される値であり、予めファイルに格納された MAC と、チェック時の生成 MAC との比較により、データ改竄の有無を判定する。

## 【 0 2 8 7 】

例えば、図 4 2 に示す A T R A C K 3 データファイルに格納される MAC は、その MAC による改竄チェック対象データが、「 I N F - s e q # 」からの複数のデータ項目に設定され、予めそれらの MAC 対象データ項目に基づいて生成された MAC がファイルに格納されることになる。すなわち、MAC ( I N F - s e q # || A || L T || … ) である。( ) 内のデータが MAC の対象、すなわち、改竄の有無の判定対象となるデータである。

## 【 0 2 8 8 】

しかしながら、A T R A C K 3 データファイル中には、様々な情報データが格納され、改竄チェック対象データが増加する場合がある。このような増加したチェック対象データも含めて新たな MAC を生成し、これを拡張 MAC としてファイル中に格納するとともに、従来の改竄チェック対象データのみを対象として生成されるオリジナル MAC については、基本的に改竄チェック対象領域を不変として設定した構成について説明する。

## 【 0 2 8 9 】

図 4 2 には、先に説明した I N F - s e q # 以下のデータを改竄チェック対象データとして設定して、生成されるオリジナル MAC 7 0 1 が A T R A C K 3 データファイルに格納されている。

## 【 0 2 9 0 】

さらに、A T R A C K 3 データファイル中の I N F スペースに記録されるいくつかの情報中に、改竄チェックの対象とすべきデータが存在する場合、オリジナル MAC 7 0 1 の MAC 生成対象データを構成するデータ、ここでは、[ I N F

－seq #」を含めて、その他のINFスペース内の改竄チェック対象データに基づいて新たなMACを生成し、これを拡張MACとしてデータファイル中に格納する。

#### 【0291】

図42において、拡張MAC [MAC (INF)] 702は、MAC (INF－seq#||path||MAC (profile) ||0thers…)によって生成される、このように、拡張MACは、オリジナルMACのMAC生成対象データの一部を含み、その他の改竄チェック対象と併せたデータに基づいて生成される。

#### 【0292】

また、拡張MACの書き換え時、すなわち、拡張MACの対象データ、すなわちINF領域の[path]以下のデータの書き換えにより、新たな拡張MACを、その書き換えデータに基づいて再生成して再格納する処理を実行する際には、拡張MACに含まれ、かつオリジナルMACの対象データでもある[INF－seq #]の書き換えを行なって新たな拡張MACの生成、格納処理を実行する。

#### 【0293】

この場合、オリジナルMACについても、その対象データである[INF－seq #]の書き換えが実行されているので、新たにオリジナルMACの計算を実行する。すなわち、拡張MACの更新時には、オリジナルMACの再生成、再格納処理を併せて実行する。

#### 【0294】

[INF－seq #]の書き換えは、例えば新たな乱数の発生による書き換え処理、あるいは、INF－seq #データのインクリメント処理等によって実行可能である。

#### 【0295】

このように、改竄チェック対象データの増加に対応して生成される拡張MACのMAC生成対象データに、オリジナルMACのMAC対象データの一部を含めて、双方のMACの共通するMAC対象データを存在させ、拡張MACの更新時には、オリジナルMACの再生成も併せて実行する構成としたので、オリジナル

MACのMAC対象データ領域を広げることなく、新たな改竄チェック用データである例えばINF内のデータの書き換え処理を常にオリジナルMACに反映させることが可能となる。

【0296】

〔記憶装置および再生装置間におけるEKB処理〕

次に、前述のツリー構造の鍵配信システムを適用した有効化キープロック（EKB）を用いて、暗号化コンテンツの復号処理に適用するコンテンツキーを取得する具体的処理構成について説明する。

【0297】

図43にATRACK3データ等の暗号化コンテンツを格納した例えばメモリスティック等の記憶装置100と、コンテンツ再生を実行する再生装置A200、再生装置B300を示す。

【0298】

記憶装置100には、暗号化コンテンツとして、図21等を用いて説明したATRACK3データファイルが格納され、再生装置においてコンテンツを再生するためには、コンテンツの復号に必要なコンテンツキーKconを取得することが必要となる。

【0299】

まず、再生装置が記憶装置からコンテンツキーを直接取得する処理態様について、図43に示す記憶装置800と再生装置A810とで説明する。まず、記憶装置800と、再生装置A810は、認証処理機能を実行する相互の制御モジュール801、811間において相互認証処理を実行する。相互認証は、例えば先に説明した図8の共通鍵暗号方式、あるいは公開鍵暗号方式による相互認証処理として実行する。この場合、記憶装置800と、再生装置A810は、それぞれの制御処理モジュール801、811が認証処理実行アルゴリズムを有し、さらに、認証処理に必要な鍵を格納していることが必要である。

【0300】

記憶装置800は、再生装置A810との相互認証の成立後、記憶装置800内の制御モジュール801において、フラッシュメモリ802に格納したATR

A C K 3 データファイルから、記憶装置のストレージキー  $K_{stm}$  で暗号化されたコンテンツキー： $E(K_{stm}, K_{con})$  または、先に説明した E K B ファイルの処理によって取得可能なキー暗号キー ( $K_{EK}$ ) で暗号化されたコンテンツキー： $E(K_{EK}, K_{con})$  のいずれかを取り出し、復号処理を実行して、コンテンツキー  $K_{con}$  を取得する。

## 【 0 3 0 1 】

記憶装置 8 0 0 は、再生装置 A 8 1 0 との相互認証時に生成したセッションキー  $K_{ses}$  を用いてコンテンツキー  $K_{con}$  の再暗号化を実行し、生成した暗号化データ： $E(K_{ses}, K_{con})$  を再生装置 A 8 1 0 に送付する。再生装置 A 8 1 0 は、制御モジュール 8 1 1 において、受領した暗号化コンテンツキー  $E(K_{ses}, K_{con})$  をセッションキー  $K_{ses}$  で復号してコンテンツキーを取得する。

## 【 0 3 0 2 】

以上、説明した手法が、記憶装置側において、コンテンツキーを復号して取り出して、これを再度セッションキーで暗号化して再生装置に送付する手法である。

## 【 0 3 0 3 】

次に、記憶装置側では復号処理を実行せず、再生装置側においてコンテンツキーを取得する処理を実行形態について説明する。

## 【 0 3 0 4 】

この処理形態を図 4 3 の記憶装置 8 0 0 と再生装置 B 8 3 0 との間の処理として説明する。記憶装置 8 0 0 は、A T R A C K 3 データファイル中の有効化キーブロック (E K B) バージョン (またはジェネレーション) から、コンテンツキーの取得に必要な対応有効化キーブロック (E K B) を特定し、特定された E K B を再生装置 B 8 3 0 に送付する。

## 【 0 3 0 5 】

再生装置 B 8 3 0 は、記憶装置から E K B を受領し、予め再生装置内のメモリ、例えば E 2 P R O M (e x. フラッシュメモリ) 内に格納したデバイスキーブロック (D K B) を用いて受領 E K B の処理を実行し、キー暗号キー ( $K_{EK}$ )

を取得する。

#### 【0306】

ここで、デバイスキーブロック (DKB) について説明する。図44を用いてデバイスキーブロック (DKB) の構成を説明する。前述したように、コンテンツ再生装置等の各デバイスは、図44 (a) に示すツリー構造の鍵配信構成の末端すなわちリーフから上位のルートに連なる各ノードのキーを有する。例えば図44 (a) に示す末端ノードのセット5 (SET5) に対応するデバイスは、リーフキーとしてのK101, ノードキーとしてK10, K1から、ルートキーKrootに至るキーセット、または、サブカテゴリーノードキーに至るキーセット、あるいはカテゴリーノードに至るキーセットを保有する。

#### 【0307】

これらの各キーは、デバイスにおいて暗号化されてデバイス内のメモリ、例えばE2PROMに格納される。このような各デバイスに保存されるリーフから特定ノード (ex. サブカテゴリーノード) またはルートまでのキーに対応するキーセットの暗号化キーセットがデバイスキーブロック (DKB) である。

#### 【0308】

デバイスキーブロック (DKB) のデータ構成例を図44 (b) に示す。図44 (b) に示すように、DKBはノードキー、およびルートキーをリーフキーで暗号化したデータと、リーフキーをデバイス (ex. 再生装置) のストレージキー: Kstdで暗号化したデータを有する暗号化キーブロックとして構成される。デバイス (ex. 再生装置) は、このデバイスキーブロック (DKB) 中のEnc (Kstd, Kleaf) を、自身のストレージキー: Kstdを用いて復号し、リーフキーKleafを取得し、さらに、取得したリーフキーKleafを用いて高位の暗号化ノードキー、暗号化ルートキーを直接復号することが可能となり、DKBの下位キーから順次復号して上位キーを取得していく処理の省略が可能となる。なおデバイスキーブロック (DKB) には、リーフの識別子であるリーフIDを含む。

#### 【0309】

デバイス固有のストレージキーは、各セット (デバイス) 毎に異なる鍵であり

、予めデバイス中のセキュアメモリ（ex. SAM）中に格納するか、あるいはリーフIDに基づいて求めることの可能な構成としてもよい。すなわち、デバイスの制御モジュール（暗号処理部）において、リーフIDに基づいて生成する構成としてもよい。具体的には、所定のセット単位で共通に格納されたマスターキー  $K_{mas}$  に基づいてリーフIDに対するハッシュを適用し、 $K_{std} = hash(K_{mas}, \text{リーフID})$  として求める構成としてもよい。

## 【0310】

図43に戻ってコンテンツキーの取得処理の説明を続ける。記憶装置800から有効化キーブロック（EKB）を受領した再生装置B830は、制御モジュール831において、メモリ832に格納したデバイスキーブロック（DKB）の復号によって得られるノードキー、ルートキー等を適用してEKBにより暗号化されたキー暗号化キー（KEK）を取得する。EKBの処理手法は、先に図5あるいは図9を用いて説明したと同様の手法である。

## 【0311】

再生装置B830は、有効化キーブロック（EKB）の処理によって取得したキー暗号化キー（KEK）を用い、さらに、記憶装置800から受領した暗号化コンテンツキー： $E(KEK, K_{con})$ の復号処理を実行してコンテンツキーを取得する。

## 【0312】

なお、図43の再生装置B830のメモリ（E2PROM）832に格納されたイニシャルEKBは、デバイス（再生装置B830）に当初から格納される簡略化したEKBファイルであり、例えば、前述の図11を用いた説明中に記載したカテゴリーノードにおいて、1つのカテゴリーノード（例えばカテゴリ＝メモリスティック）の下位に接続されるリーフに対応するデバイスに共通に格納される暗号化キーブロックである。

## 【0313】

例えばカテゴリーノードの持つキーがK01であればK01で暗号化されたルートキー： $Enc(K01, K_{root})$ がイニシャルEKBとして格納される。デバイスはイニシャルEKBの処理によりルートキーを取得することが可能と

なり、例えばルートキーによって暗号化されたキー暗号化キー（KEK）を格納したEKBを受領した場合には、イニシャルEKBから得たルートキーを用いてキー暗号化キー（KEK）を取得することが可能となる。

## 【0314】

なお、イニシャルEKBは、1つのカテゴリーノードに属するデバイスに共通に提供する構成とする態様に限らず、複数のカテゴリーノードに共通に構成してもよい。例えばメモリスティックのカテゴリーノードのノードキーK01、コンテンツ再生機能を持つPCのカテゴリーノードのノードキーをK10、ネットワーク対応の形態再生装置のカテゴリーノードのノードキーをK11としたとき、これらの各デバイスに予め、Enc(K01, Kroot)、Enc(K10, Kroot)、Enc(K11, Kroot)の3種類の暗号化ルートキーを格納したイニシャルEKBを設定して出荷することにより、それぞれの異なるデバイスにおいて共通に利用可能な暗号化コンテンツの配信を行なうことが可能となる。

## 【0315】

図45に、再生装置のメモリ（ex. E2PROM）にデバイスキーブロック（DKB）と、イニシャルEKBとして自己録音、自己再生用の有効化キーブロック（EKB）を格納した構成例を示す。また、図46にこれらのキーブロックを利用したコンテンツキーの取得処理例を示す。

## 【0316】

図45の構成について説明する。デバイス（ex. 記録再生器）は、図45（a）のリーフに対応するデバイスであり、ツリー構成の第8段目に構成されるカテゴリーノードKn8のカテゴリーに属するデバイスである。デバイスには、（b）に示すEnc(Kstd, Kleaf)～Enc(Kleaf, Kn8)のデバイスキーブロック（DKB）が格納される。この構成は、先に説明したDKBと同様であるが、リーフキーによって直接暗号化されて格納されたデータは、リーフキーの直上のノードキーKn47からカテゴリーノードキーであるKn8までのキーとして構成される。

## 【0317】

さらに、デバイスは、自己録音、再生用の有効化キーブロック（EKB）を保

有し、自己のデバイスでのコンテンツ録音、再生時には、この自己録音、再生用の有効化キープブロック（EKB）とデバイスキープブロック（DKB）との処理によりコンテンツキーKconを取得して、コンテンツの復号、暗号化を実行する。

#### 【0318】

図46に、図45（b）のDKB，EKBを持つデバイスにおけるコンテンツキーの取得処理において実行するステップを示す。まずステップS4601において、デバイスは、リーフIDに基づいてストレージキーKstdを抽出する。ストレージキーKstdは、リーフIDに基づいてデバイス中のセキュアメモリから抽出するか、あるいは前述したように、マスターキーKmasとリーフIDに基づいて算出する。

#### 【0319】

次に、S4602において、ストレージキーKstdに基づいてデバイスキープブロック（DKB）の処理、すなわちEnc（Kstd，Kleaf）の復号を実行し、リーフキーを求める。次に、S4603において、リーフキーKleafに基づいてデバイスキープブロック（DKB）の処理、すなわちEnc（Kleaf，Kn8）の復号を実行し、カテゴリリーノードキーを求める。DKBは、リーフキーにより直接暗号化されたノードキーが格納されているので、高位のノードキーを直接リーフキーによる復号処理によって取得することが可能となる。

#### 【0320】

次に、ステップS4604において、ノードキーKn8からEKB処理を実行し、順次高位のノードキーを求めて、最上位キーであるルートキーを算出する。次に、ステップS4605において、有効化キープブロック（EKB）の処理によって求めたルートキーKrootを用いてEnc（Kroot，KEK）の復号処理を実行してキー暗号化キーKEKを求める。最後にステップS4606において、取得したキー暗号化キーKEKを用いて、コンテンツデータに付随したデータ中に格納されたEnc（KEK，Kcon）の復号処理を実行してコンテンツキーKconを取得する。

#### 【0321】

図 4 5 (b) に示す有効化キープブロック (E K B) は自己録再用の E K B であるが、様々なコンテンツをデバイスにダウンロードする際に、そのコンテンツに対応する E K B を併せてダウンロードし、コンテンツに対応付けて E K B をメモリに格納し、コンテンツの再生時にダウンロードしたコンテンツ対応の E K B に対して図 4 6 の処理を実行することも可能である。また、図 4 5 (b) に示すデバイスキープブロック (D K B) は、上位から 8 段のノード  $K_n 8$  のノードキーまでを直接リーフキーで暗号化したデータを D K B の暗号化キーデータとした構成であるが、格納するノードキーは、さらに上位、あるいは下位までのノードキーとしてもよい。

#### 【 0 3 2 2 】

以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲の欄を参酌すべきである。

#### 【 0 3 2 3 】

##### 【発明の効果】

以上、説明したように、本発明のデータ処理装置、および方法によれば、記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行する構成において、検証値をコンテンツのカテゴリ毎に独立した検証値として生成して格納する構成としたので、コンテンツデータの検証処理の効率化、さらに検証後の記録デバイスに対するダウンロード処理、あるいは検証後の再生処理等を効率的に実行することが可能となる。

#### 【 0 3 2 4 】

さらに、本発明のデータ処理装置、および方法によれば、カテゴリは、コンテンツの種類、あるいは、コンテンツの暗号処理鍵として設定されるコンテンツキー  $K_{con}$  を暗号化して提供する有効化キープブロック (E K B) の管理エンティティに基づいて設定された構成としたので、例えば有効化キープブロック (E K B

) の管理エンティティ別に、コンテンツデータの検証処理を独立して実行可能となり、処理の効率化が実現される。

【 0 3 2 5 】

さらに、本発明のデータ処理装置、および方法によれば、コンテンツまたはヘッダ情報中の異なるデータ領域をMAC生成対象として複数のメッセージ認証符号(MAC)を生成するとともに、複数のメッセージ認証符号(MAC)の各MAC生成対象データ領域の一部を共通データとし、いずれかのMAC更新時に、前記共通データを更新し、他方のMAC更新処理も併せて実行する構成としたので、オリジナルMACのMAC対象データ領域を広げることなく、新たな拡張MACの改竄チェック用データの書き換え処理を常にオリジナルMACに反映させることが可能となる。

【図面の簡単な説明】

【図 1】

本発明のデータ処理装置の使用概念を説明する図である。

【図 2】

本発明のデータ処理装置のシステム構成例およびデータ経路例を示す図である。

【図 3】

本発明のデータ処理装置における各種キー、データの暗号化処理について説明するツリー構成図である。

【図 4】

本発明のデータ処理装置における各種キー、データの配布に使用される有効化キーブロック(EKB)の例を示す図である。

【図 5】

本発明のデータ処理装置におけるコンテンツキーの有効化キーブロック(EKB)を使用した配布例と復号処理例を示す図である。

【図 6】

本発明のデータ処理装置における有効化キーブロック(EKB)のフォーマット例を示す図である。

【図 7】

本発明のデータ処理装置における有効化キーブロック（E K B）のタグの構成を説明する図である。

【図 8】

本発明のデータ処理装置における有効化キーブロック（E K B）と、コンテンツキー、コンテンツを併せて配信するデータ構成例を示す図である。

【図 9】

本発明のデータ処理装置における有効化キーブロック（E K B）と、コンテンツキー、コンテンツを併せて配信した場合のデバイスでの処理例を示す図である。

【図 1 0】

本発明のデータ処理装置における有効化キーブロック（E K B）とコンテンツを記録媒体に格納した場合の対応について説明する図である。

【図 1 1】

本発明のデータ処理装置における階層ツリー構造のカテゴリ分類の例を説明する図である。

【図 1 2】

本発明のデータ処理装置における簡略化有効化キーブロック（E K B）の生成過程を説明する図である。

【図 1 3】

本発明のデータ処理装置における有効化キーブロック（E K B）の生成過程を説明する図である。

【図 1 4】

本発明のデータ処理装置における簡略化有効化キーブロック（E K B）を説明する図である。

【図 1 5】

本発明のデータ処理装置における再生装置と記憶装置の構成を示すブロック図である。

【図 1 6】

本発明のデータ処理装置における記憶装置内の記憶ユニットに記憶されているデータを説明する図である。

【図 1 7】

本発明のデータ処理装置における記憶装置のフラッシュメモリに記憶されるデータを説明するための図である。

【図 1 8】

本発明のデータ処理装置における再生管理ファイルのデータ構成を概略的に示す図である。

【図 1 9】

本発明のデータ処理装置におけるデータファイルのデータ構成を概略的に示す図である。

【図 2 0】

本発明のデータ処理装置における再生管理ファイルのデータ構成をより詳細に示す図である。

【図 2 1】

本発明のデータ処理装置におけるデータファイルのデータ構成をより詳細に示す図である。

【図 2 2】

本発明のデータ処理装置におけるデータファイルの属性ヘッダの一部を示す図である。

【図 2 3】

本発明のデータ処理装置におけるデータファイルの属性ヘッダの一部を示す図である。

【図 2 4】

本発明のデータ処理装置におけるモードの種類と、各モードにおける録音時間等を示す図である。

【図 2 5】

本発明のデータ処理装置におけるコピー制御情報を説明するための図である。

【図 2 6】

本発明のデータ処理装置におけるデータファイルの属性ヘッダの一部を示す図である。

【図 2 7】

本発明のデータ処理装置におけるデータファイルの各データブロックのヘッダを示す略線図である。

【図 2 8】

本発明のデータ処理装置におけるデータ記録処理フローを示す図である。

【図 2 9】

本発明のデータ処理装置において適用可能な相互認証処理を示す図である。

【図 3 0】

本発明のデータ処理装置におけるデータ再生処理フローを示す図である。

【図 3 1】

本発明のデータ処理装置における配信鍵許可情報ファイルのフォーマットを示す図である。

【図 3 2】

本発明のデータ処理装置におけるデータ格納態様を示す図である。

【図 3 3】

本発明のデータ処理装置におけるキー有効化ブロック（E K B）を使用したデータ復号処理フローを示す図である。

【図 3 4】

本発明のデータ処理装置における有効化キーブロック（E K B）と、認証キーを併せて配信するデータ構成と、デバイスでの処理例を示す図（その 1）である。

【図 3 5】

本発明のデータ処理装置における有効化キーブロック（E K B）と、認証キーを併せて配信するデータ構成と、デバイスでの処理例を示す図（その 2）である。

【図 3 6】

本発明のデータ処理装置における仮想メモリカードを適用した認証処理シー

ケンスを示す図である。

【図 3 7】

本発明のデータ処理装置において適用可能なインテグリティ・チェック値（ICV）の生成に使用するMAC値生成例を示す図である。

【図 3 8】

本発明のデータ処理装置におけるインテグリティ・チェック値（ICV）の格納態様を説明する図である。

【図 3 9】

本発明のデータ処理装置におけるMAC値を格納するシーケンスページフォーマットを示す図である。

【図 4 0】

本発明のデータ処理装置におけるICVを格納するプールページフォーマットを示す図である。

【図 4 1】

本発明のデータ処理装置におけるICVチェック処理フローを示す図である。

【図 4 2】

本発明のデータ処理装置においてて着ようか能な拡張MACの生成、格納処理を説明する図である。

【図 4 3】

本発明のデータ処理装置におけるキー有効化ブロック（EKB）を用いたコンテンツキーの取得処理態様を説明する図である。

【図 4 4】

本発明のデータ処理装置において使用されるデバイスキーブロック（DKB）の構成について説明する図である。

【図 4 5】

本発明のデータ処理装置におけるデバイスキーブロック（DKB）、キー有効化ブロック（EKB）の格納構成例を示す図である。

【図 4 6】

本発明のデータ処理装置におけるデバイスキーブロック（DKB）、キー有効

化ブロック（E K B）を用いたコンテンツキーの取得処理態様を説明する図である。

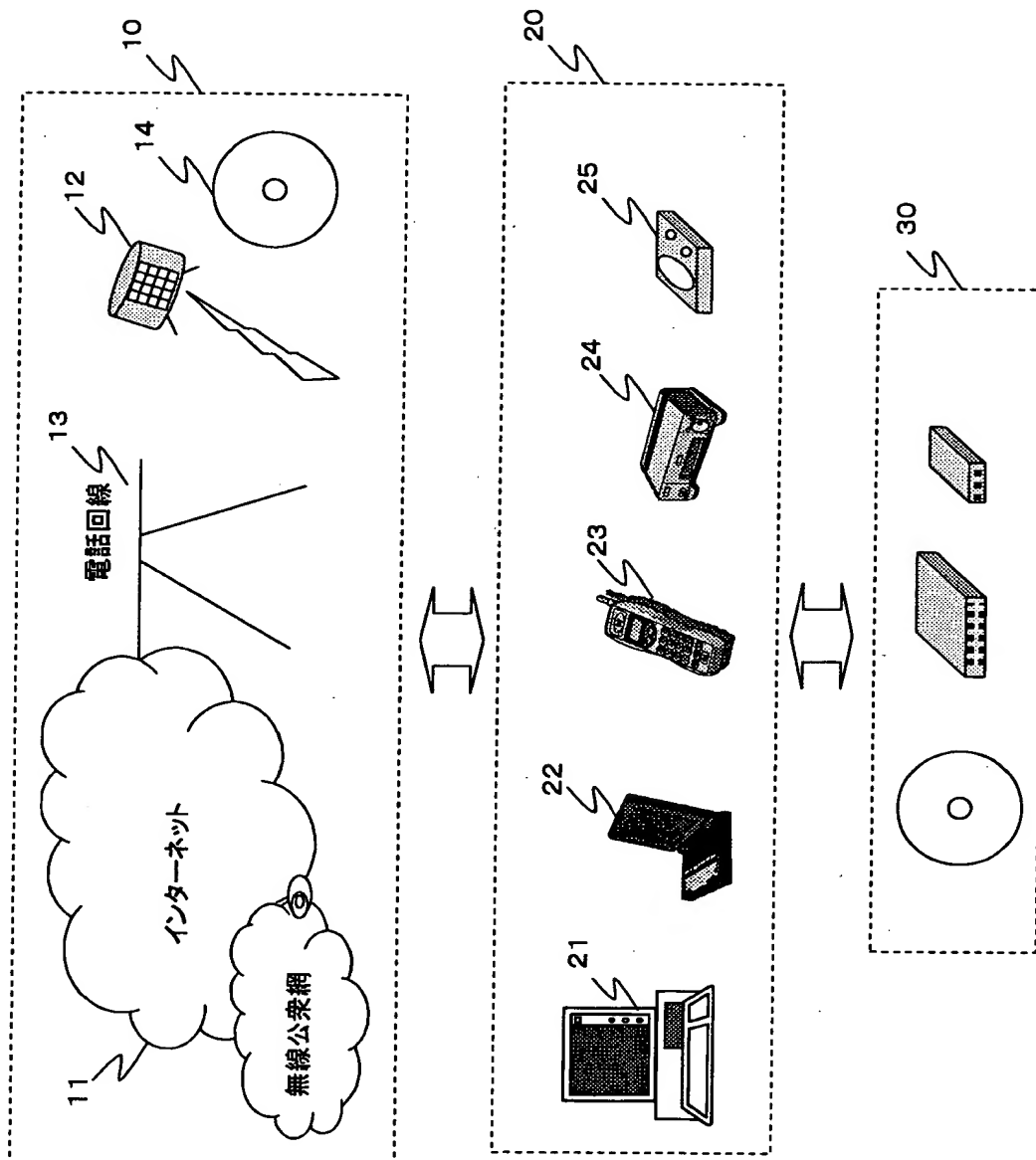
【符号の説明】

- 1 0 コンテンツ配信手段
- 1 1 インターネット
- 1 2 衛星放送
- 1 3 電話回線
- 1 4 メディア
- 2 0 データ処理手段
- 2 1 パーソナルコンピュータ（P C）
- 2 2 ポータブルデバイス（P D）
- 2 3 携帯電話、P D A
- 2 4 記録再生器、ゲーム端末
- 2 5 再生装置
- 3 0 記憶手段
- 1 0 0 パーソナルコンピュータ（P C）
- 2 0 0 再生装置
- 3 0 0 記憶装置
- 6 0 1 バージョン
- 6 0 2 デプス
- 6 0 3 データポインタ
- 6 0 4 タグポインタ
- 6 0 5 署名ポインタ
- 6 0 6 データ部
- 6 0 7 タグ部
- 6 0 8 署名
- 3 3, 4 3 制御モジュール
- 5 0, 6 0 乱数発生ユニット
- 5 1, 6 1 記憶ユニット

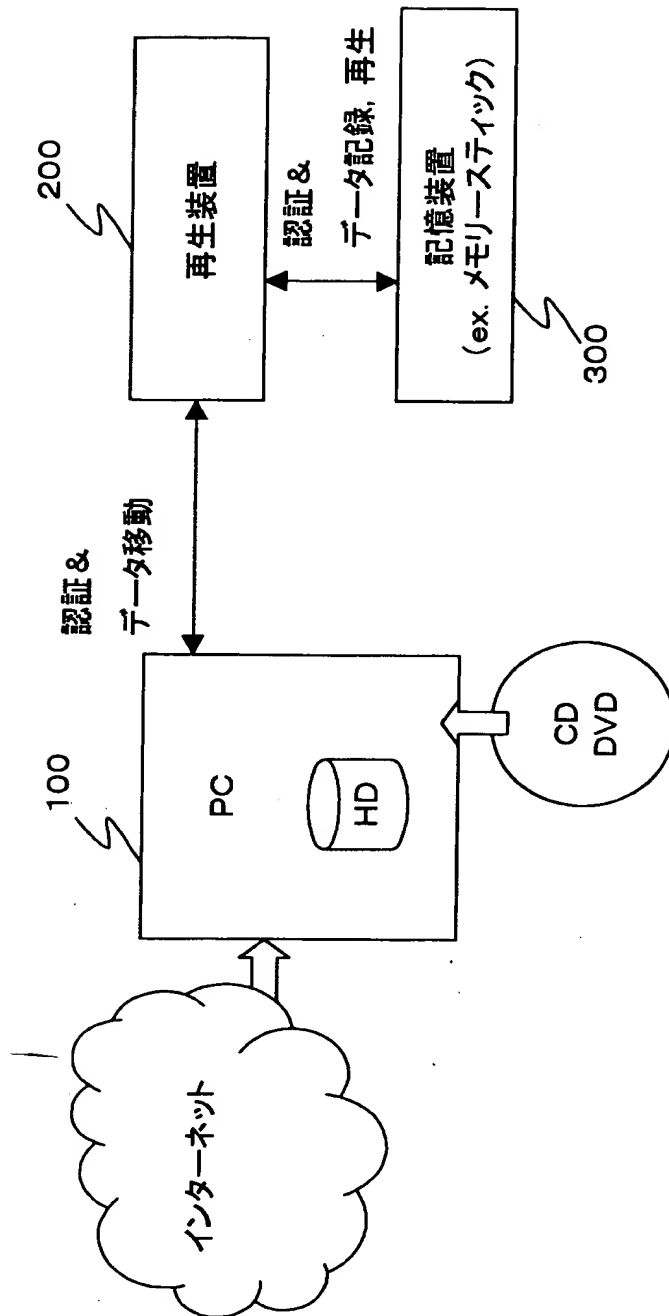
- 5 2, 6 2 鍵生成／演算ユニット
- 5 3, 6 3 相互認証ユニット
- 5 4, 7 4 暗号化／復号ユニット
- 5 5, 6 5 制御ユニット
- 3 4 フラッシュメモリ
- 4 4 編集モジュール
- 4 5 圧縮／伸長モジュール
- 4 6 スピーカ
- 4 9 メモリ
- 8 0 0 記憶装置
- 8 0 1 制御モジュール
- 8 0 2 フラッシュメモリ
- 8 1 0 再生装置 A
- 8 1 1 制御モジュール
- 8 3 0 再生装置 B
- 8 3 1 制御モジュール
- 8 3 2 メモリ

【書類名】 図面

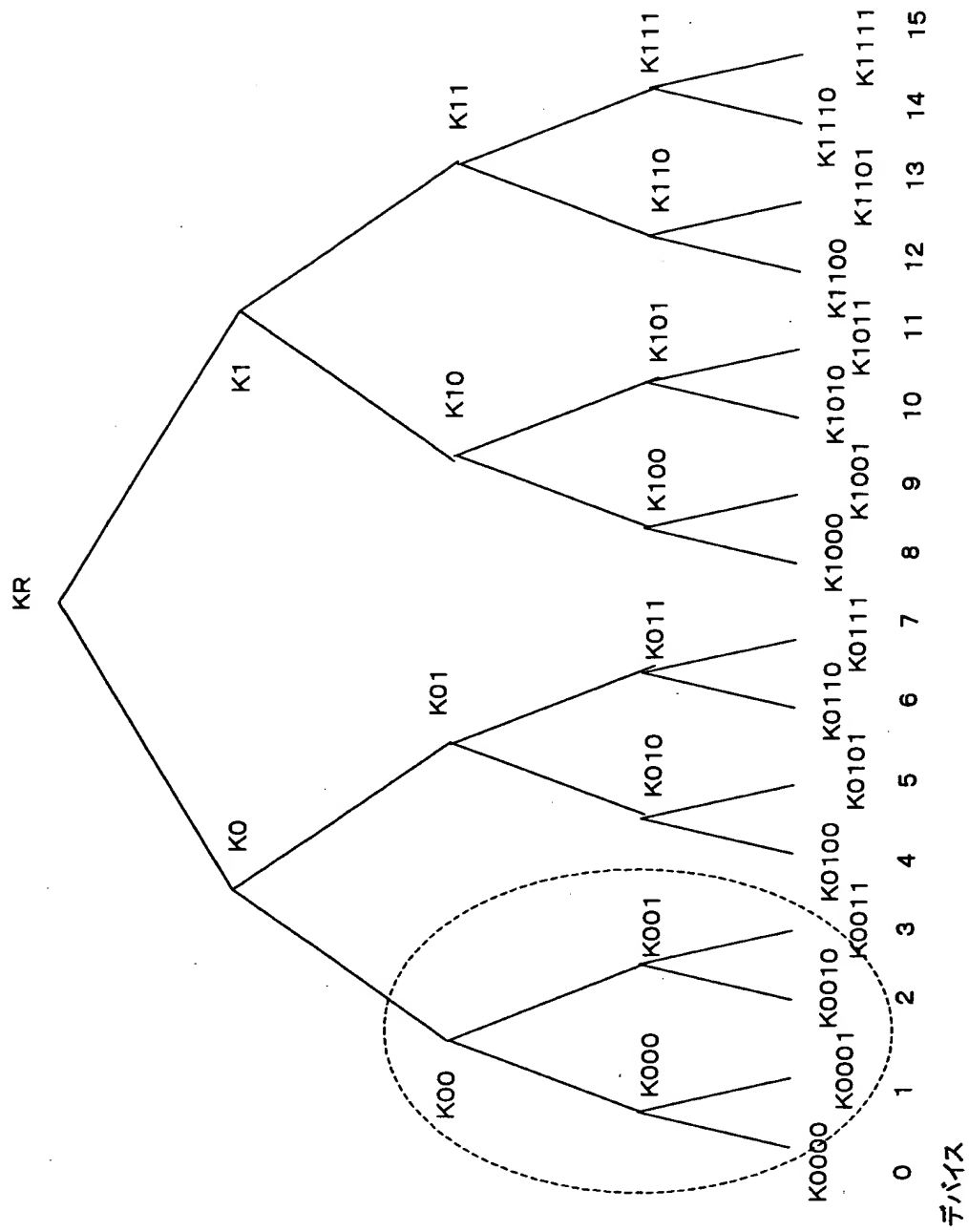
【図 1】



【図2】



【図 3】



【図 4】

## (A) 有効化キーブロック(EKB: Enabling Key Block) 例1

デバイス0, 1, 2にバージョン:tのノードキーを送付

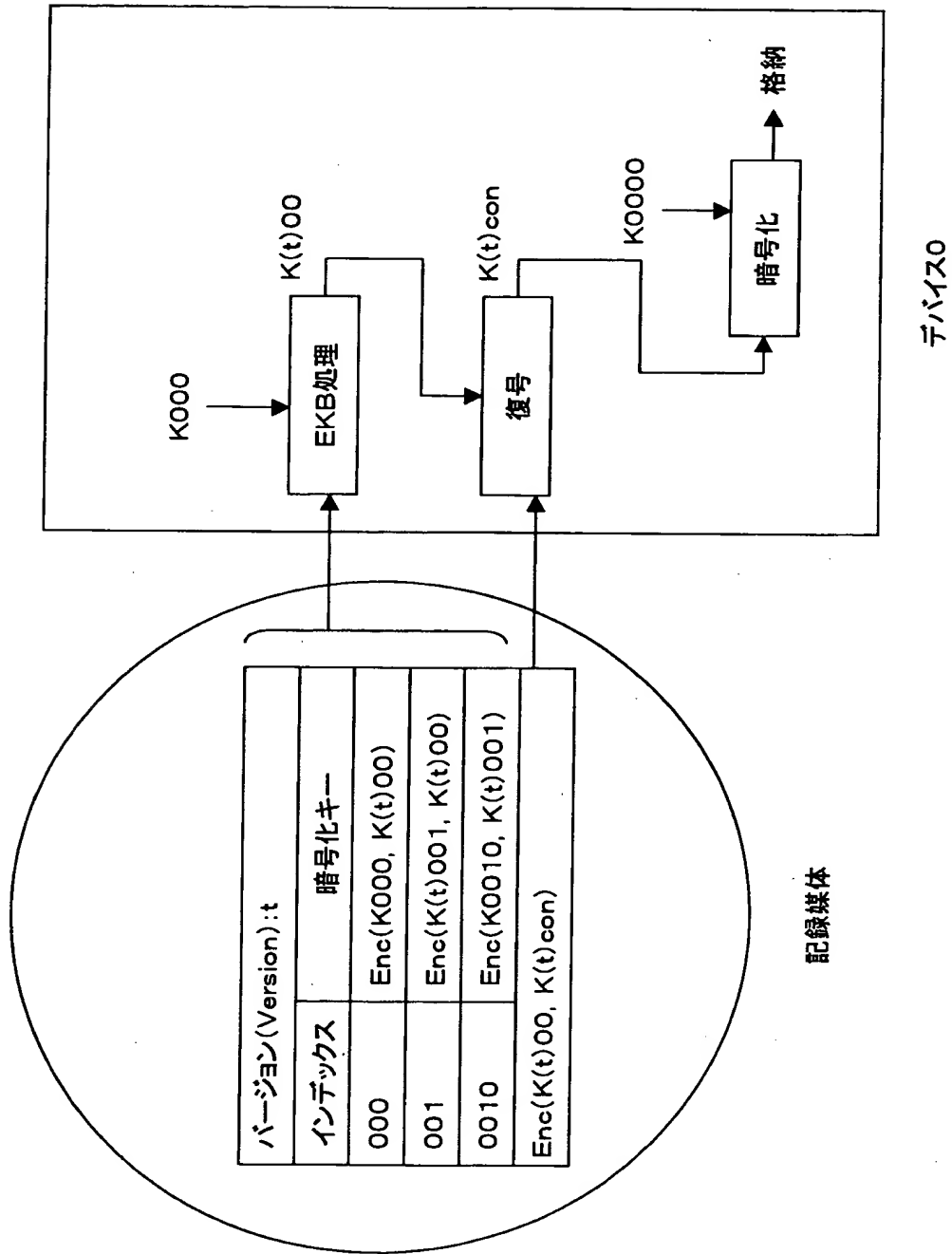
バージョン(Version):t	
インデックス	暗号化キー
0	Enc(K(t)0, K(t)R)
00	Enc(K(t)00, K(t)0)
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

## (B) 有効化キーブロック(EKB: Enabling Key Block) 例2

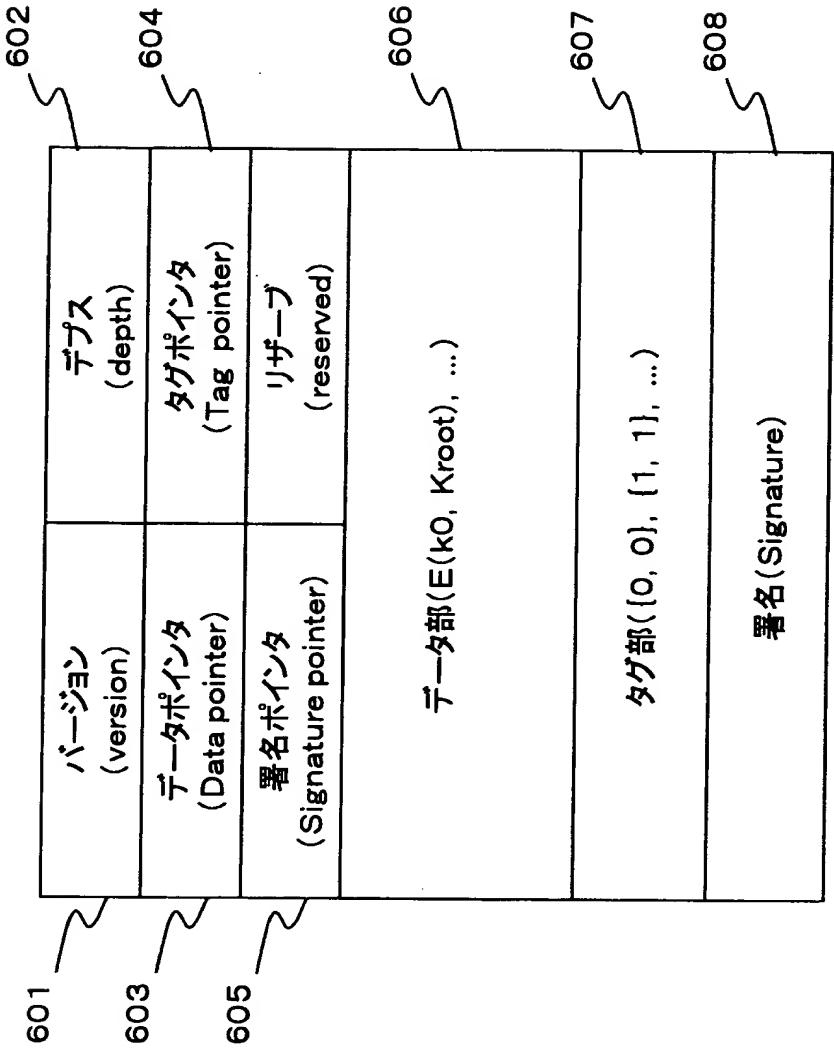
デバイス0, 1, 2にバージョン:tのノードキーを送付

バージョン(Version):t	
インデックス	暗号化キー
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

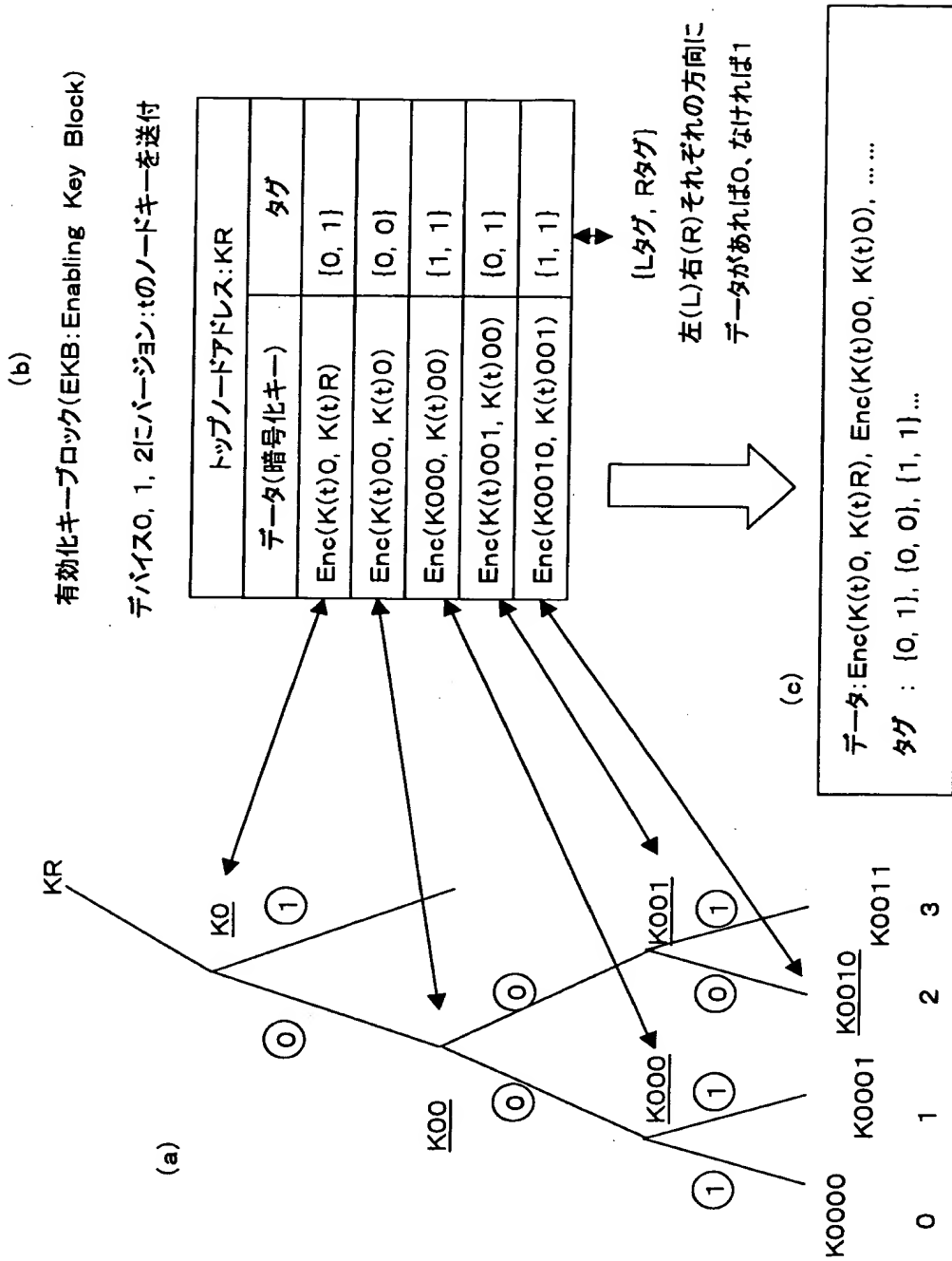
【図 5】



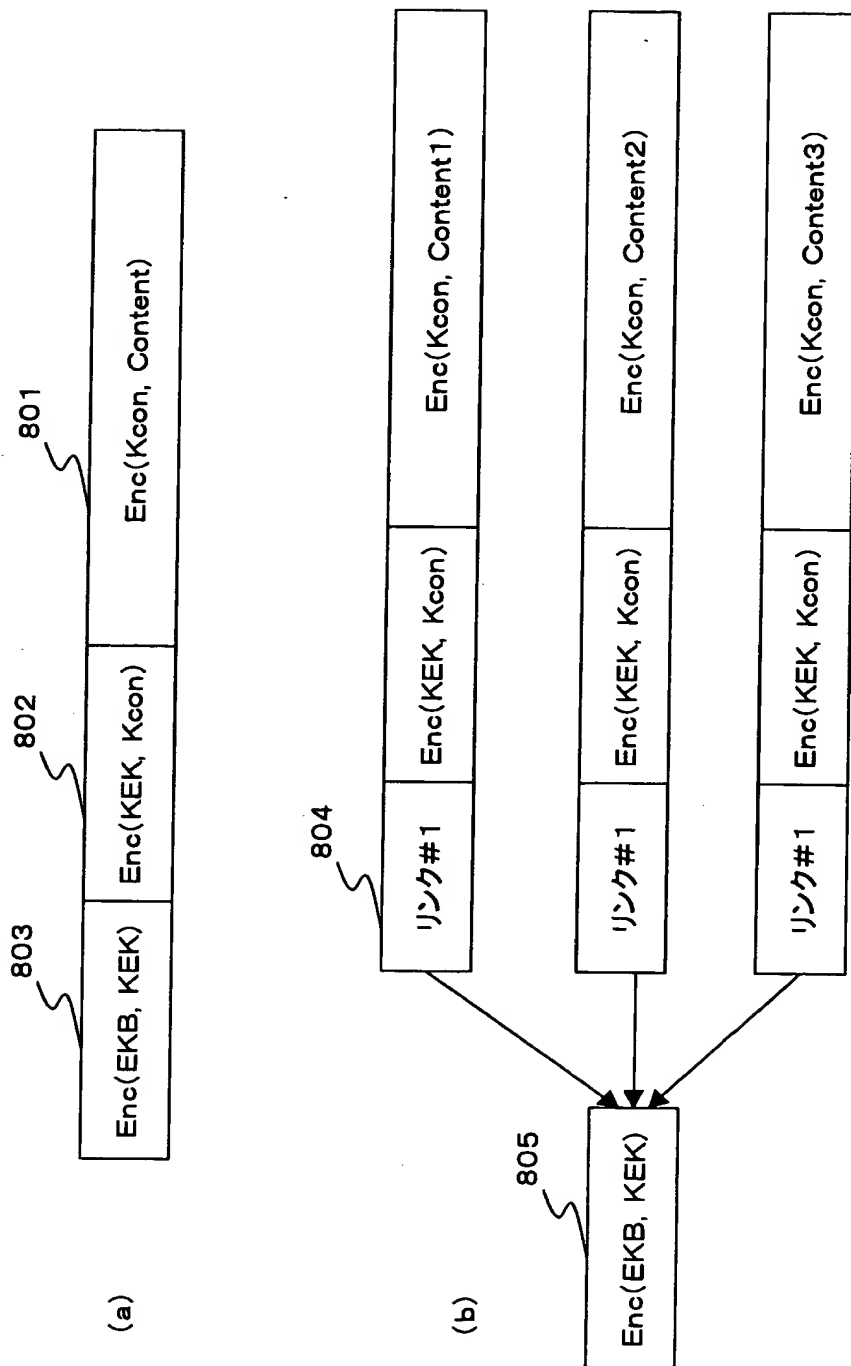
【図 6】



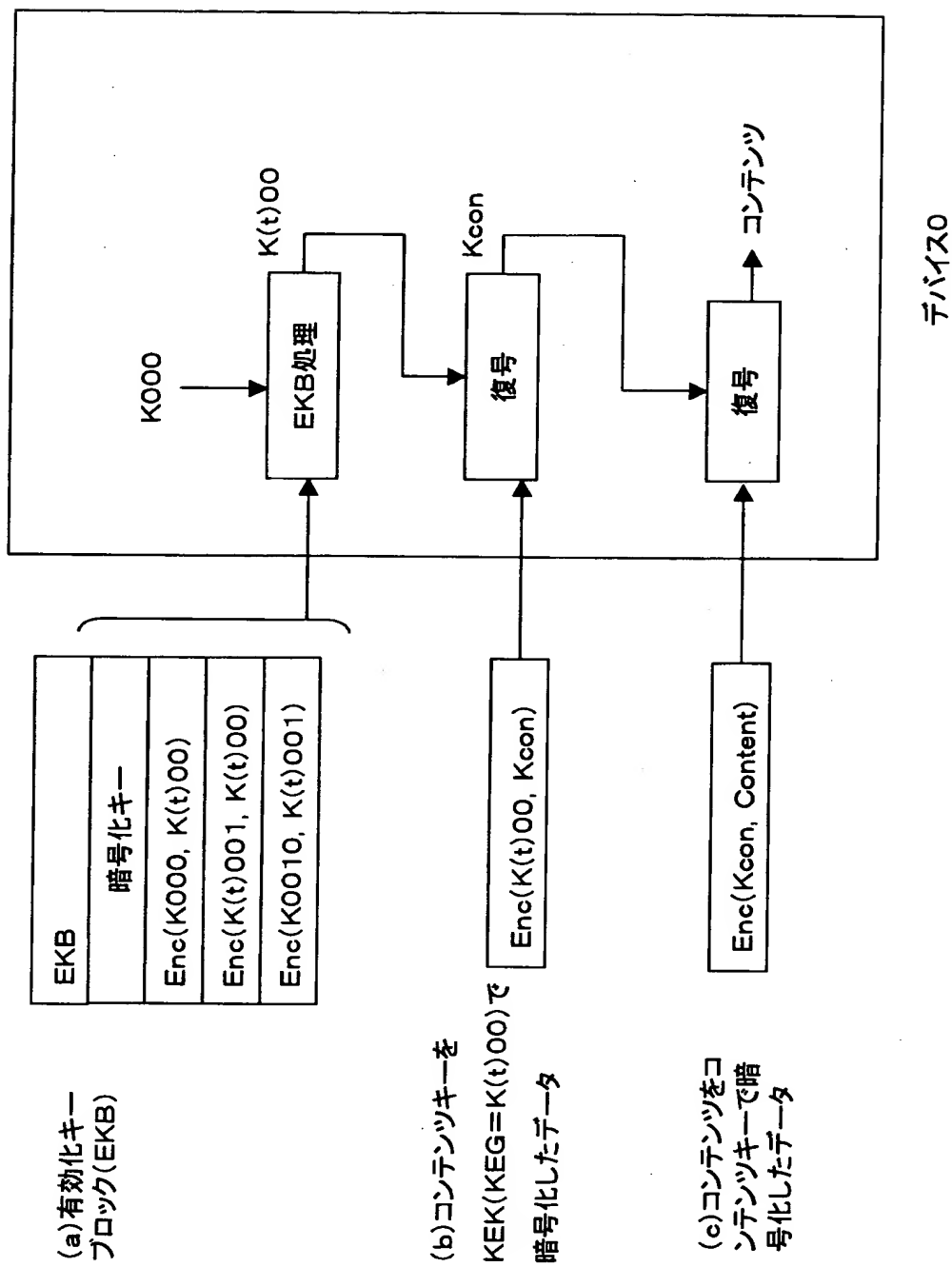
【図 7】



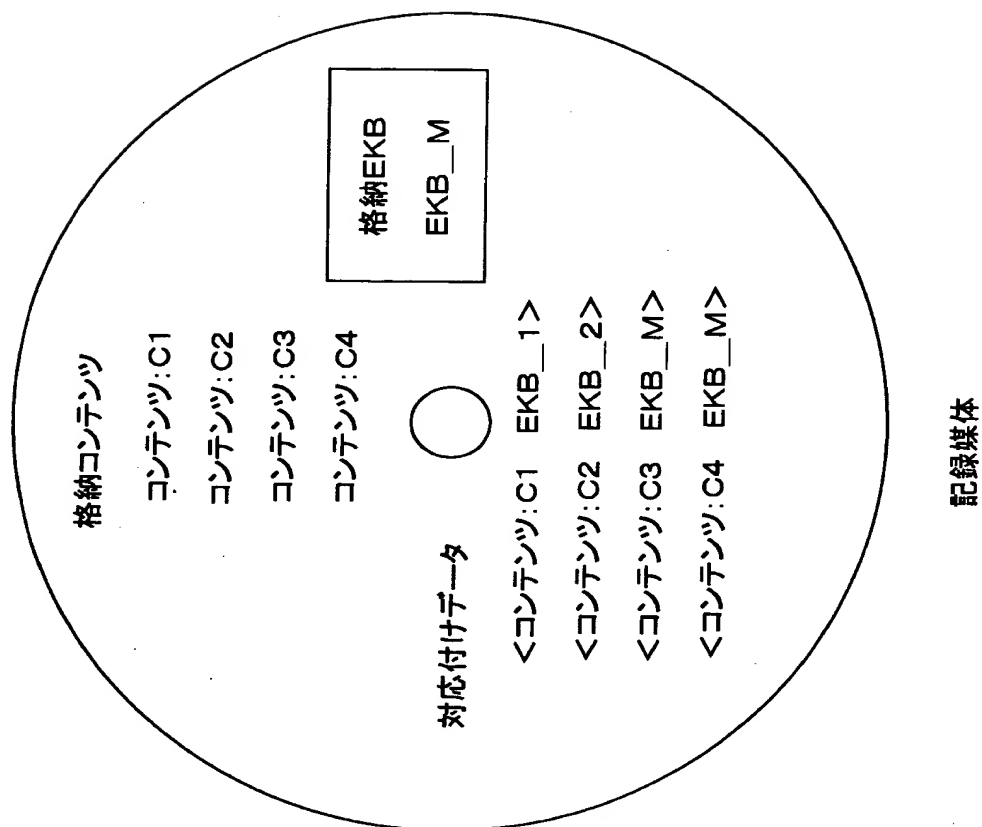
【図 8】



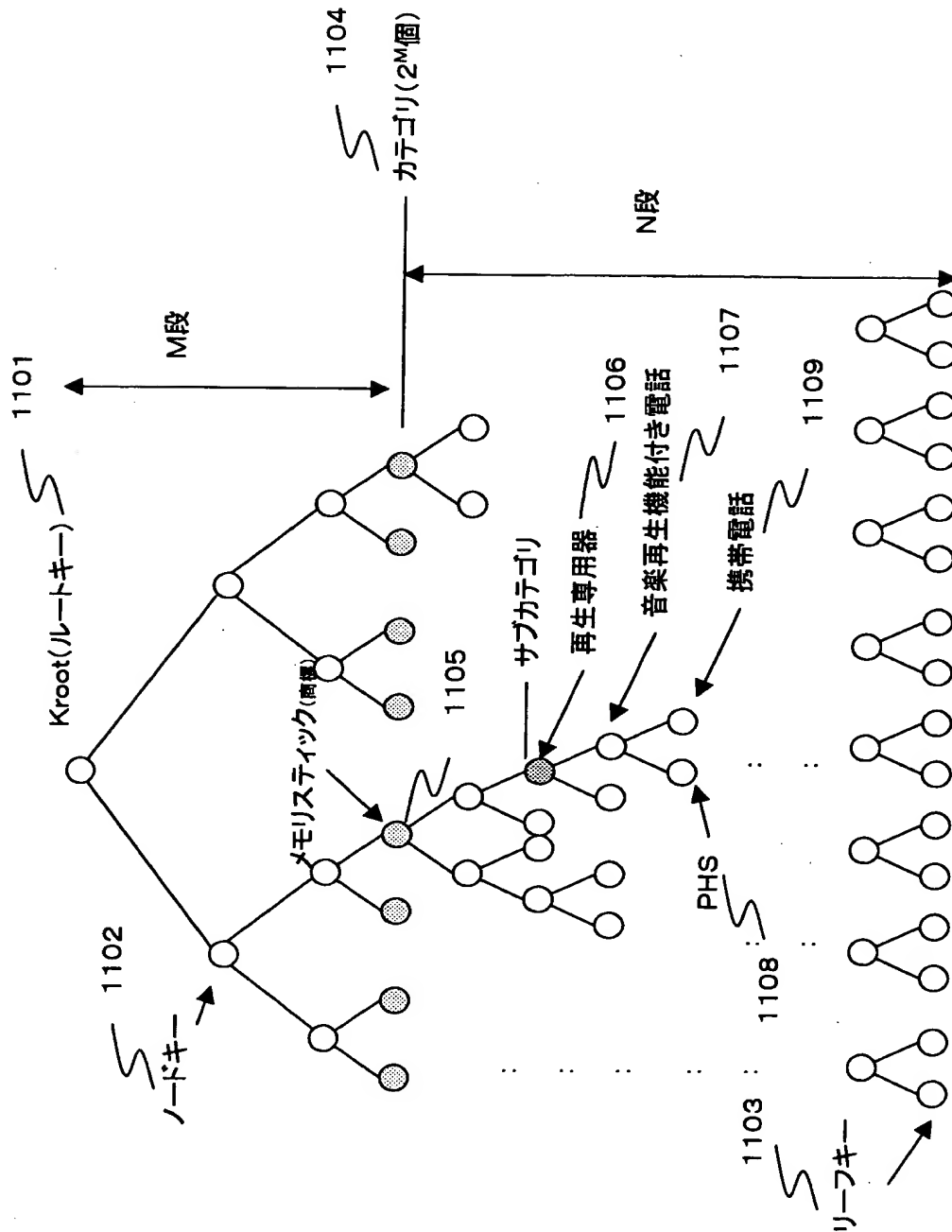
【図9】



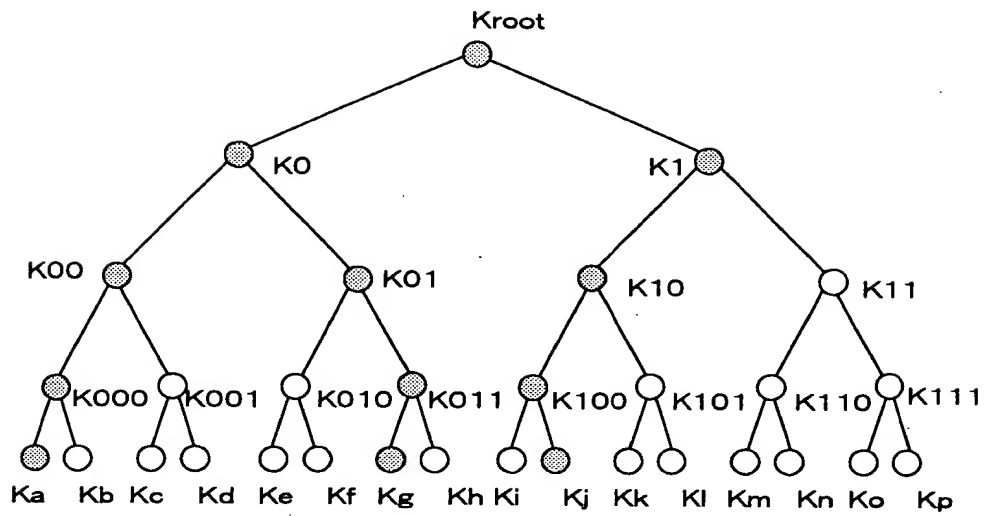
【図 1 0】



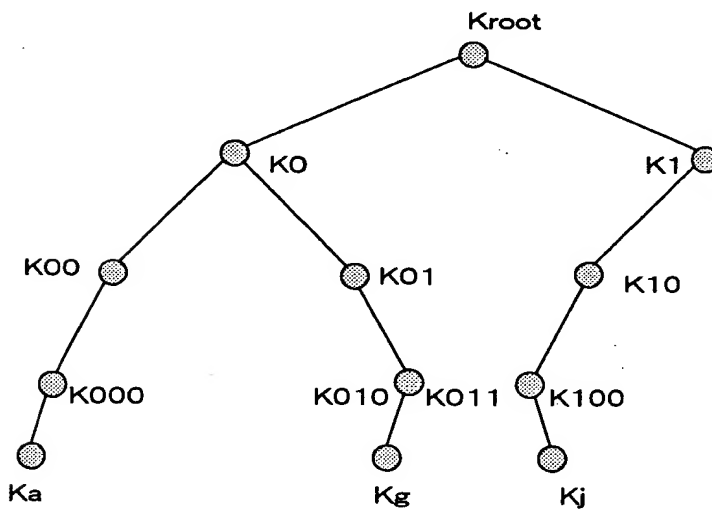
【図 11】



【図 1 2】



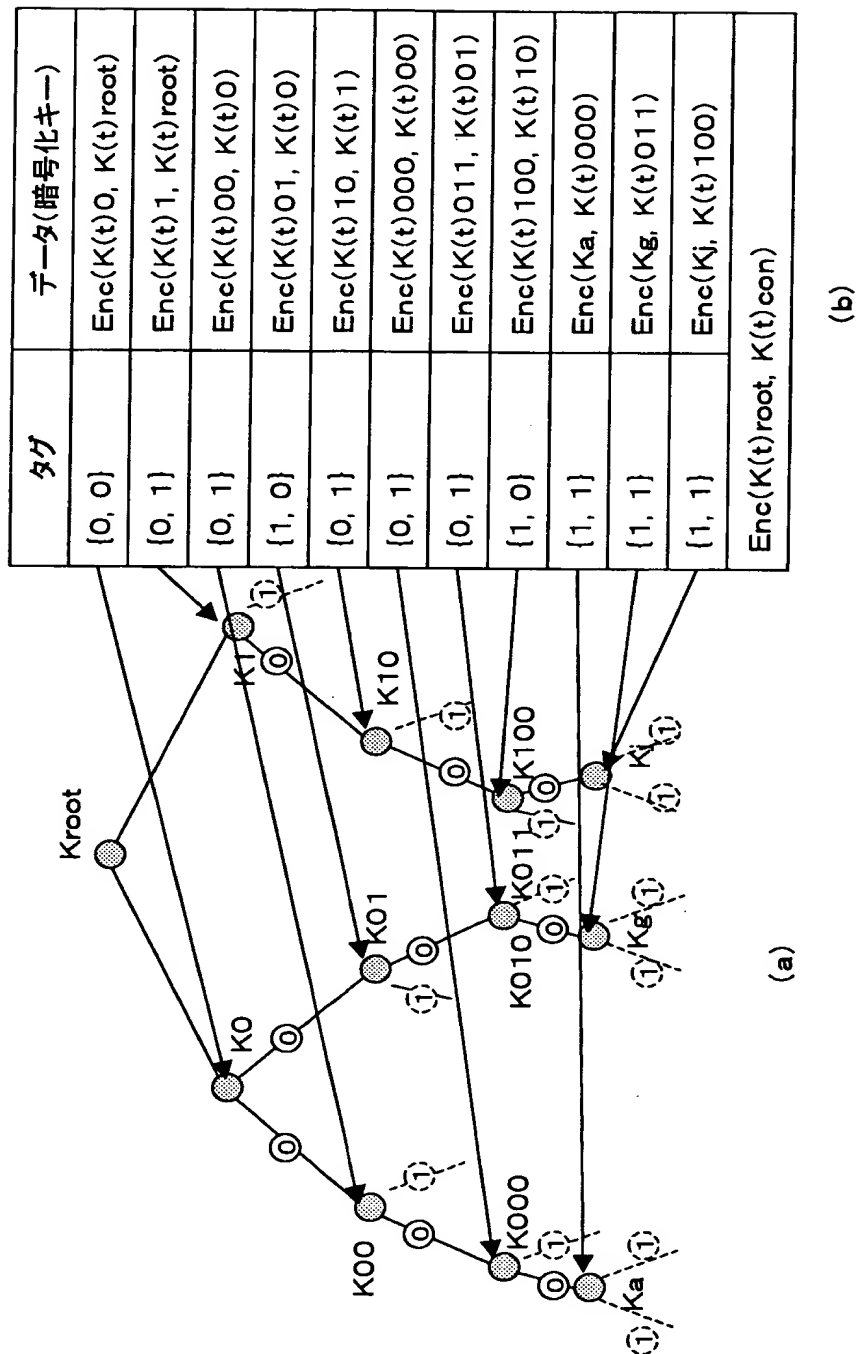
(a)



(b)

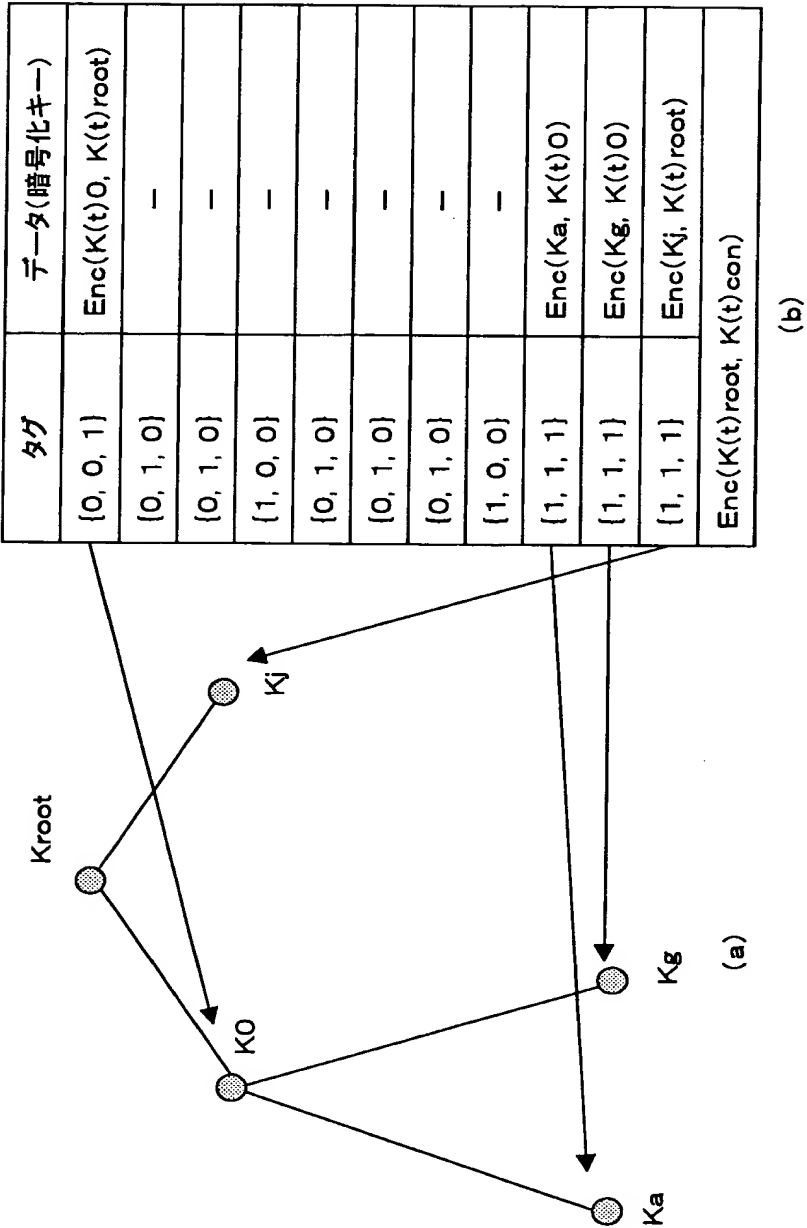
【图 13】

有効化キーブロック(EKB:Enabling Key Block)を用いた  
デバイスKa, Kg, Kjへのバージョンのコンテンツキー送付処理

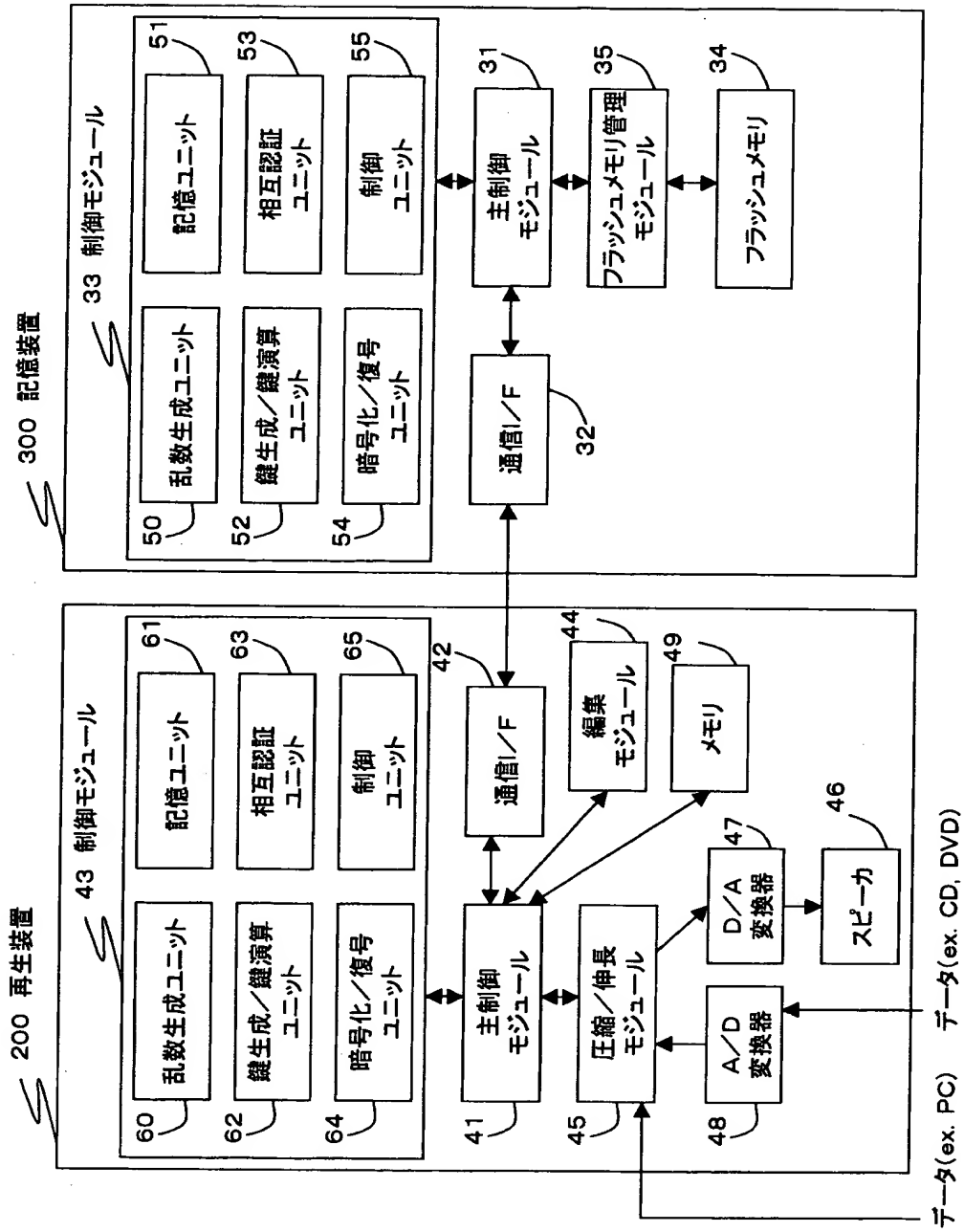


【図 1 4】

簡略化した有効化キーブロック(EKB: Enabling Key Block)を用いた  
デバイスKa, Kg, Kjへのバージョンtのコンテンツキー送付処理



【図 15】

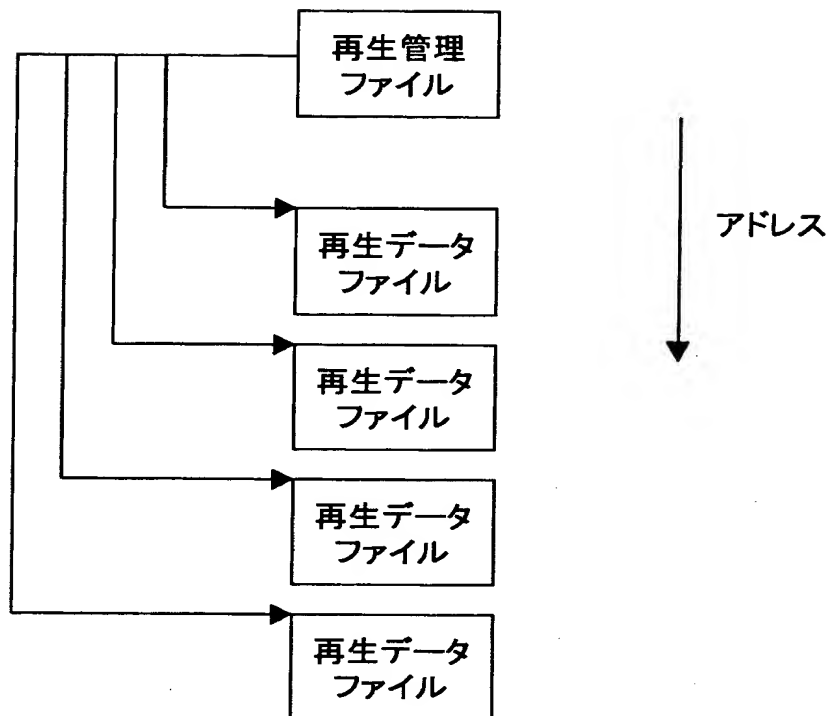


【図 1 6】

記憶装置の記憶ユニットに格納されるデータ

認証鍵データ	IK0
	IK1
	IK2
	IK3
	:
	:
	IK30
	IK31
装置識別データ	ID0
記憶用鍵データ	Kstm

【図 1 7】

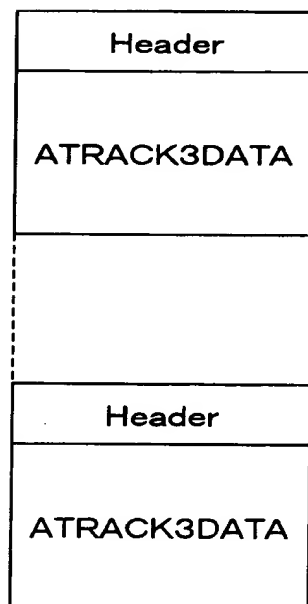
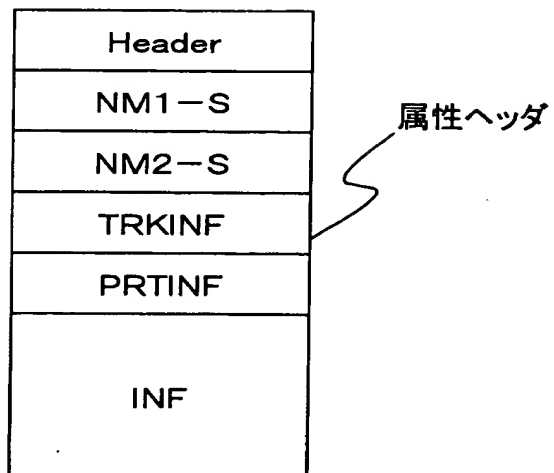


【図 1 8】

再生管理ファイル

Header
NM1-S
NM2-S
TRKTBL
INF-S

【図 1 9】



【図 20】

再生管理ファイル

A

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x0000	BLKID-TLO			Reserved		Mcode		REVISION				Reserved				
0x0010	SN1C+L		SN2C+L		SINFSIZE		T-TRK		VerNo.		Reserved					

B

0x0020	NM1-S(256)									
0x0120	NM2-S(512)									
0x0310										
0x0320	Reserved(4)		EKB version		E(Kstm, Kcon)					
0x0330	E(KEKn, Kcon)				C_MAC[0]					
0x0340	Reserved(8)				Reserved(3)		MGR	S-YMDhms		
0x0350	TRK-001	TRK-002	TRK-003	TRK-004	TRK-005	TRK-006	TRK-007	TRK-008		
0x0360	TRK-009	TRK-010	TRK-011	TRK-012	TRK-013	TRK-014	TRK-015	TRK-016		
0x0660	TRK-393	TRK-394	TRK-395	TRK-396	TRK-397	TRK-398	TRK-399	TRK-400		
0x0670	INF-S(14720)									
0x3FFF	BLKID-TLO		Reserved		Mcode		REVISION		Reserved	

C

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
INF	0X00	ID	0X00	SIZE	Mcode	C+L	Reserved	DATA可変長								

【図21】

ATRAC3データファイル

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x0000	BLKID—HD0				Reserved		Mcode		Reservrd				BLOCK SERIAL			
0x0010	N1C+L		N2C+L		INFSIZE		T—PRT		T—SU				INX		XT	
0x0020	NM1—S(256)															
0x0120	NM2—S(512)															
0x0310																
0x0320	Reserved(3)		EKI		EKB version				E(Kstm, Kcon)							
0x0330	E(KEKn, Kcon)								C_MAC[n]							
0x0340	Reserved(8)								INF_seq#			A	LT		FNo	
0x0350	MG(D)SERIAL—nnn(Upper)								MG(D)SERIAL—nnn(Lower)							
0x0360	CONNUM				YMDhms—S				YMDhms—E				XCC	CT	CC	CN
0x0370	PRTSIZE				PRTKEY								Reserved(8)			
0x0380					CONNUM0				PRTSIZE(0x0388)				PRTKEY			
0x0390					Reserved(8)								CONNUM0			
	INF(0x0400)															
0x3FFF	BLKID—HD0				Reserved		Mcode		Reservrd				BLOCK SERIAL			
0x4000	BLKID—A3D				Reserved		Mcode		CONNUM0				BLOCK SERIAL			
0x4010	BLOCKSEED								INITIALIZATION VECTOR							
0x4020	SU—000(Nbyte=384byte)															
0x41A0	SU—001(Nbyte)															
0x4320	SU—002(Nbyte)															
0x04A0	SU—041(Nbyte)															
0x7DA0	Reserved(Nbyte=208byte)															
0x7F20	BLK SEED															
0x7FF0	BLKID—A3D				Reserved		Mcode		CONNUM0				BLOCK SERIAL			

【図 2 2】

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x0000	BLKID-HD0			Reserved		Mcode		Reservrd			BLOCK SERIAL					
0x0010	N1C+L		N2C+L		INFSIZE		T-PRT		T-SU			INX		XT		
0x0020	NM1-S(256)															
0x0120	NM2-S(512)															
0x0310																

【図 2 3】

0x0320	Reserved(3)	EKI	EKB version	E(Kstm, Kcon)						
0x0330	E(KEKn, Kcon)			C_MAC[n]						
0x0340	Reserved(8)			INF_seq#		A	LT	FNo		
0x0350	MG(D) SERIAL-nnn(Upper)			MG(D) SERIAL-nnn(Lower)						
0x0360	CONNUM		YMDhms-S	YMDhms-E		XCC	CT	CC	CN	

【図 2 4】

bit7: ATRACK3のモード    0: Dual                    1: Joint

bit6, 5, 4: 3bitのNはモードの値

N	モード	時間	転送レート	SU	バイト
7	HQ	47min	176kbps	31SU	512
6		58min	146kbps	38SU	424
5	EX	64min	132kbps	42SU	384
4	SP	81min	105kbps	53SU	304
3		90min	94kbps	59SU	272
2	LP	128min	66kbps	84SU	192
1	mono	181min	47kbps	119SU	136
0	mono	258min	33kbps	169SU	96

bit3: Reserved

bit2: データ区分    0: オーディオ    1: その他

bit1: 再生SKIP    0: 通常再生    1: SKIP

bit0: エンファンス    0: OFF    1: ON(50/15  $\mu$  S)

【図 2 5】

bit7: コピー許可    0: コピー禁止    1: コピー可

bit6: 世代    0: オリジナル    1: 第1世代以上

HCMS bit5-4: 高速デジタルコピーに関するコピー制御

00: コピー禁止    01: コピー第1世代    10: コピー可  
コピー第1世代のコピーした子供はコピー禁止とする

bit3-2: MagicGate認証レベル

00: Level10(Non-MG)    01: Level1  
02: Level2    11: Reserved

Level10以外はデバインド、コンバインできません

bit1, 0: Reserved

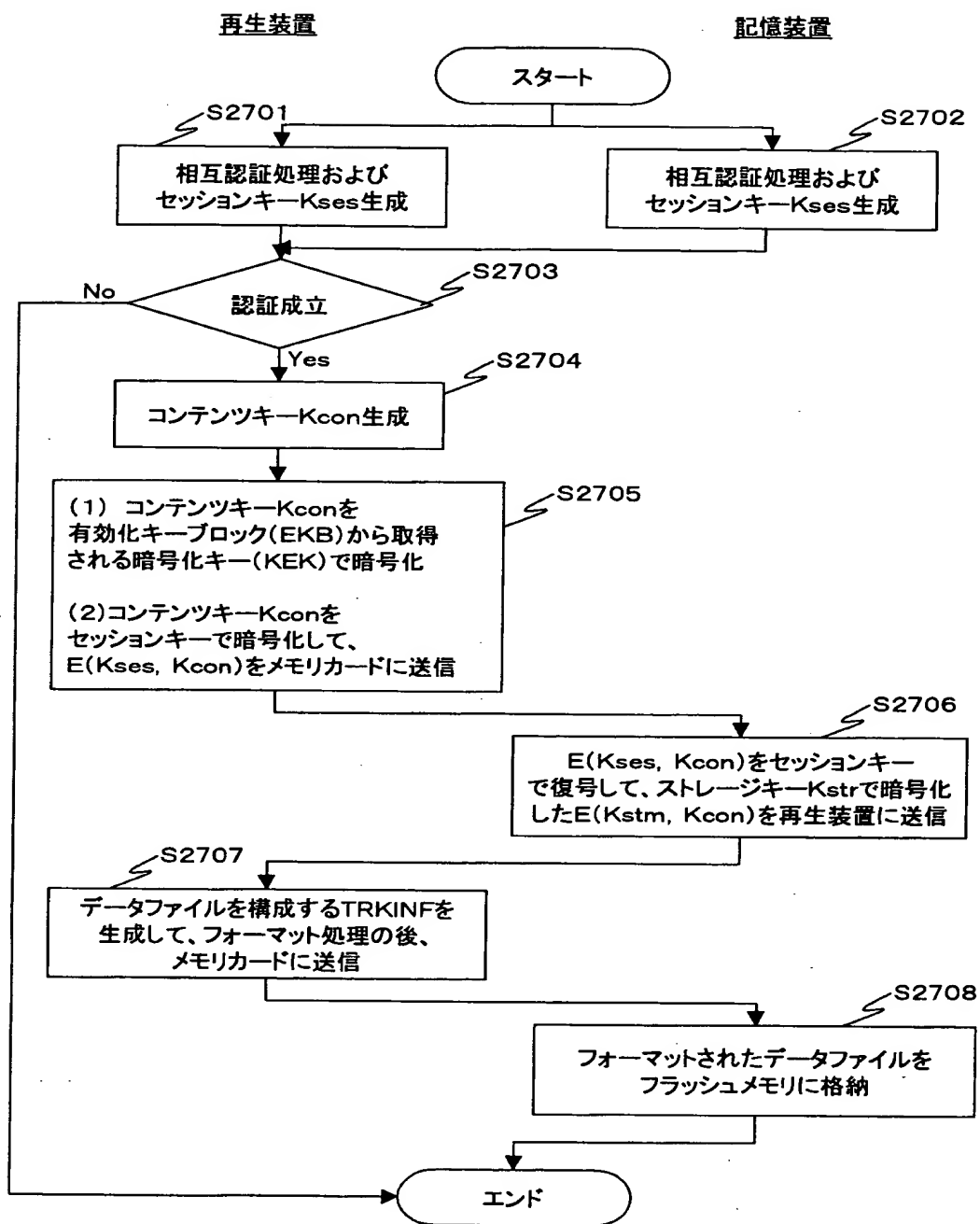
【図 2 6】

0x0370	PRTSIZE	PRTKEY		Reserved(8)
0x0380		CONNUM0	PRTSIZE(0x0388)	PRTKEY
0x0390		Reserved(8)		CONNUM0

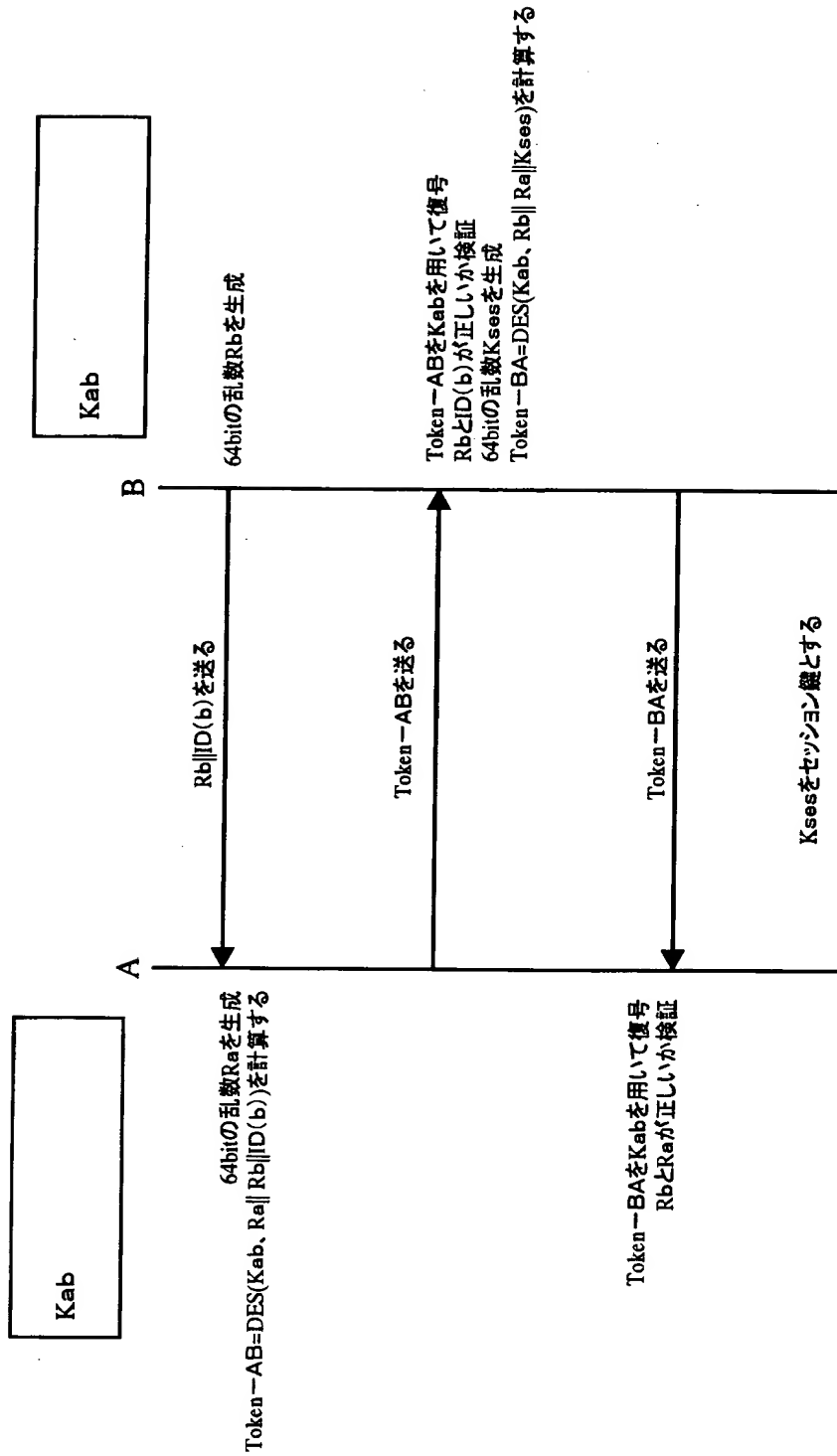
【図 2 7】

0x4000	BLKID-A3D	Reserved	Mcode	CONNUM0	BLOCK SERIAL
0x4010	BLOCKSEED			INITIALIZATION VECTOR	
0x4020	SU-000(Nbyte=384byte)				

【図 28】

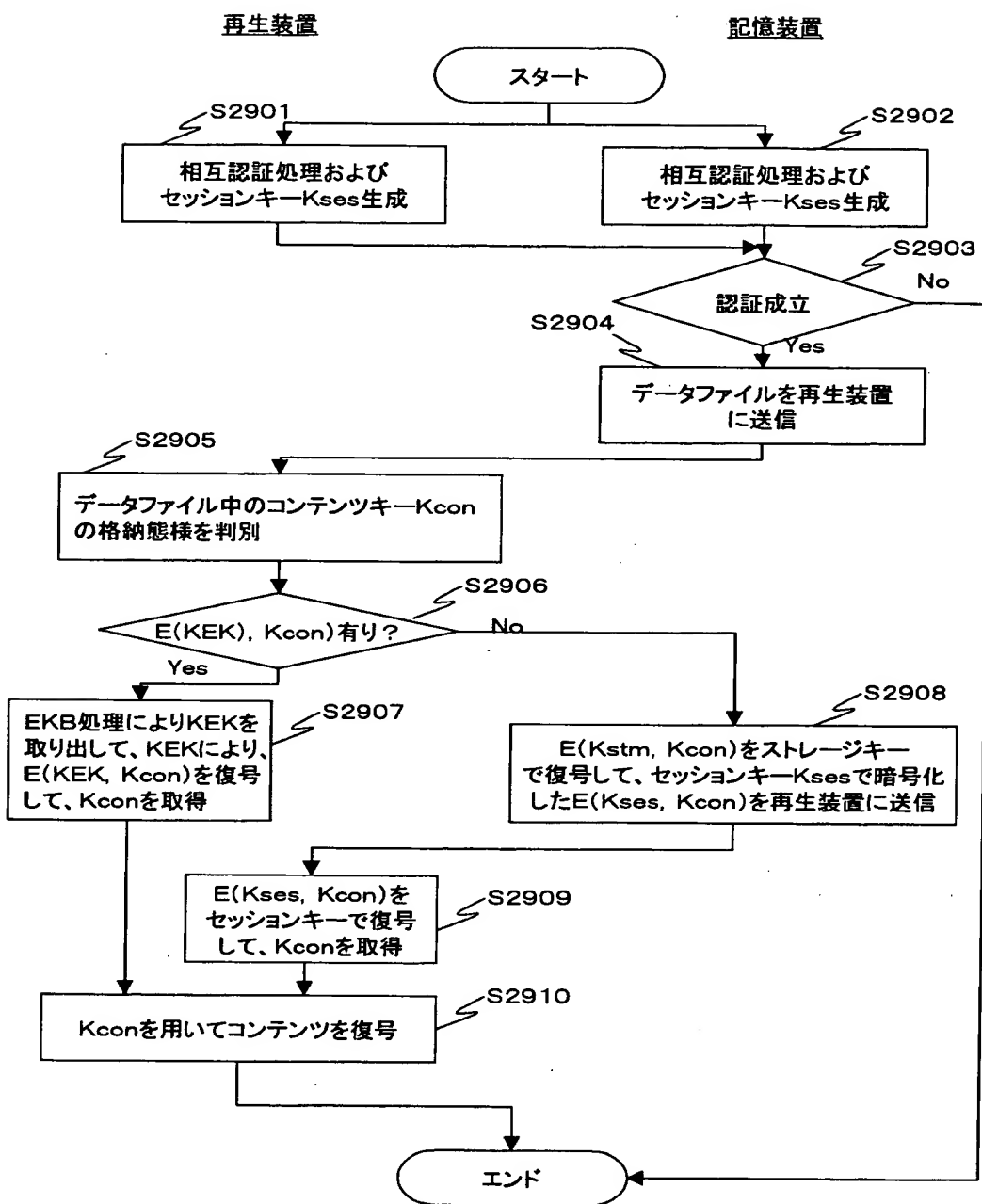


【図 2 9】



ISO/IEC 9798-2 対称鍵暗号技術を用いた相互認証および鍵共有方式

【図30】

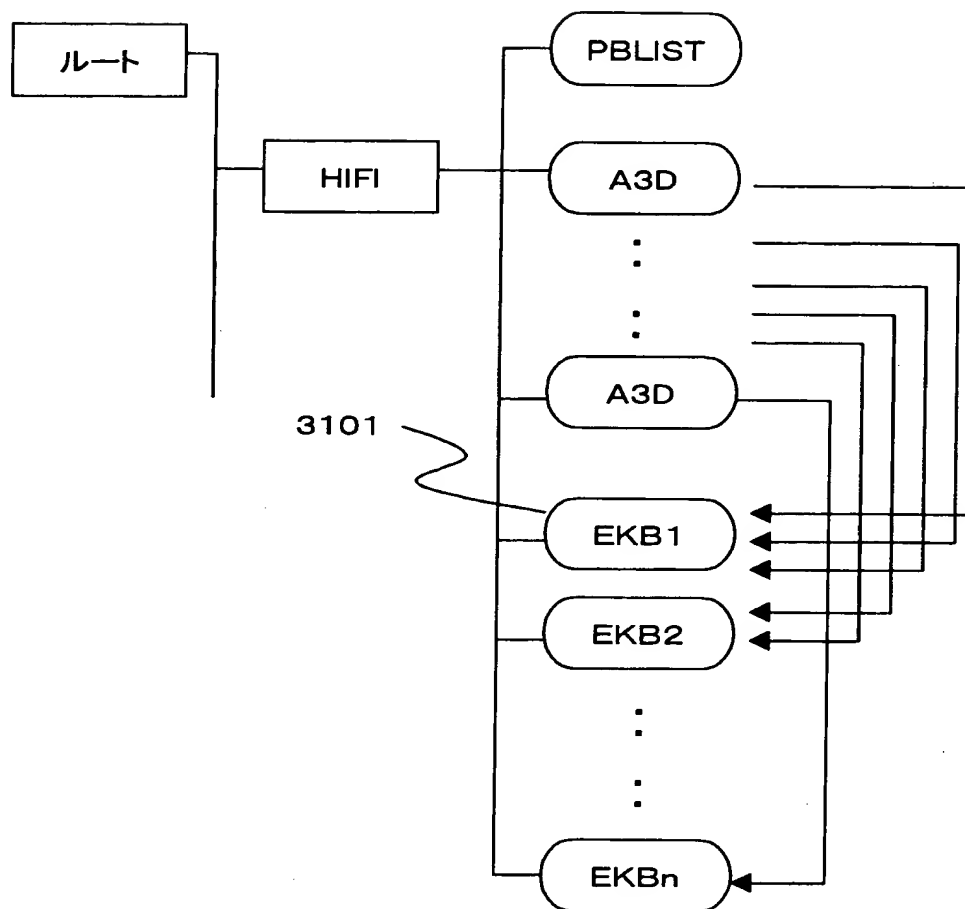


【図 3 1】

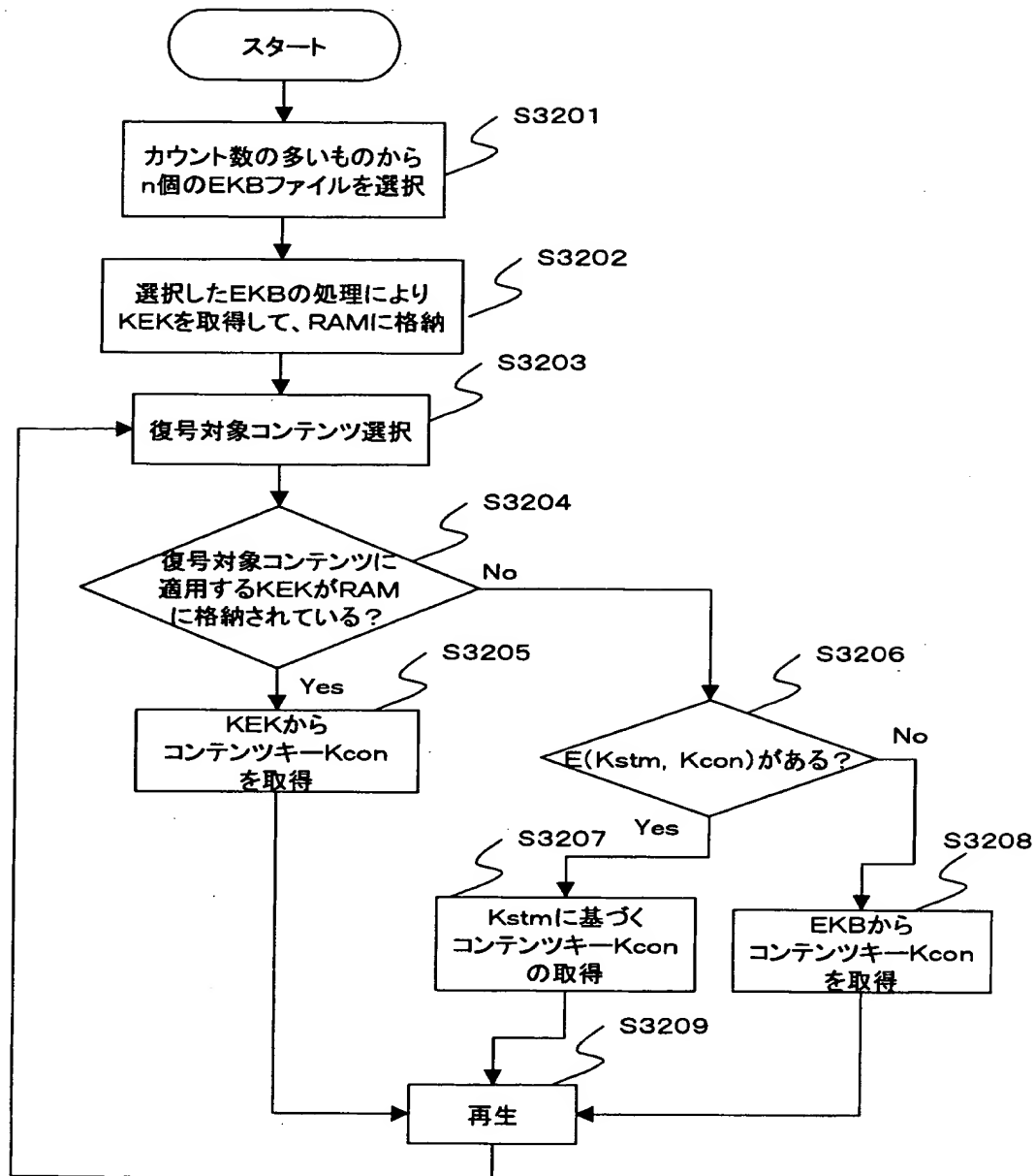
配信鍵許可情報ファイル

0x0000	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x0010	BLKID-EKB		Reserved		Mcode		Reserved(3)		LKF		Link Count					
0x0020	Version		EA		Reserved		Reserved(8)		Reserved(8)							
0x0030	Version		EA		Reserved		Reserved(8)		KEK1							
0x0040	Size of tag part		Size of key part		Size of Sign part		E(Version)									
0x0050	Tag part ({X, 0, 0}, {X, 1, 1}, ..... )															
	Fill to 64bit alignment															
	Key part															
	Signature															

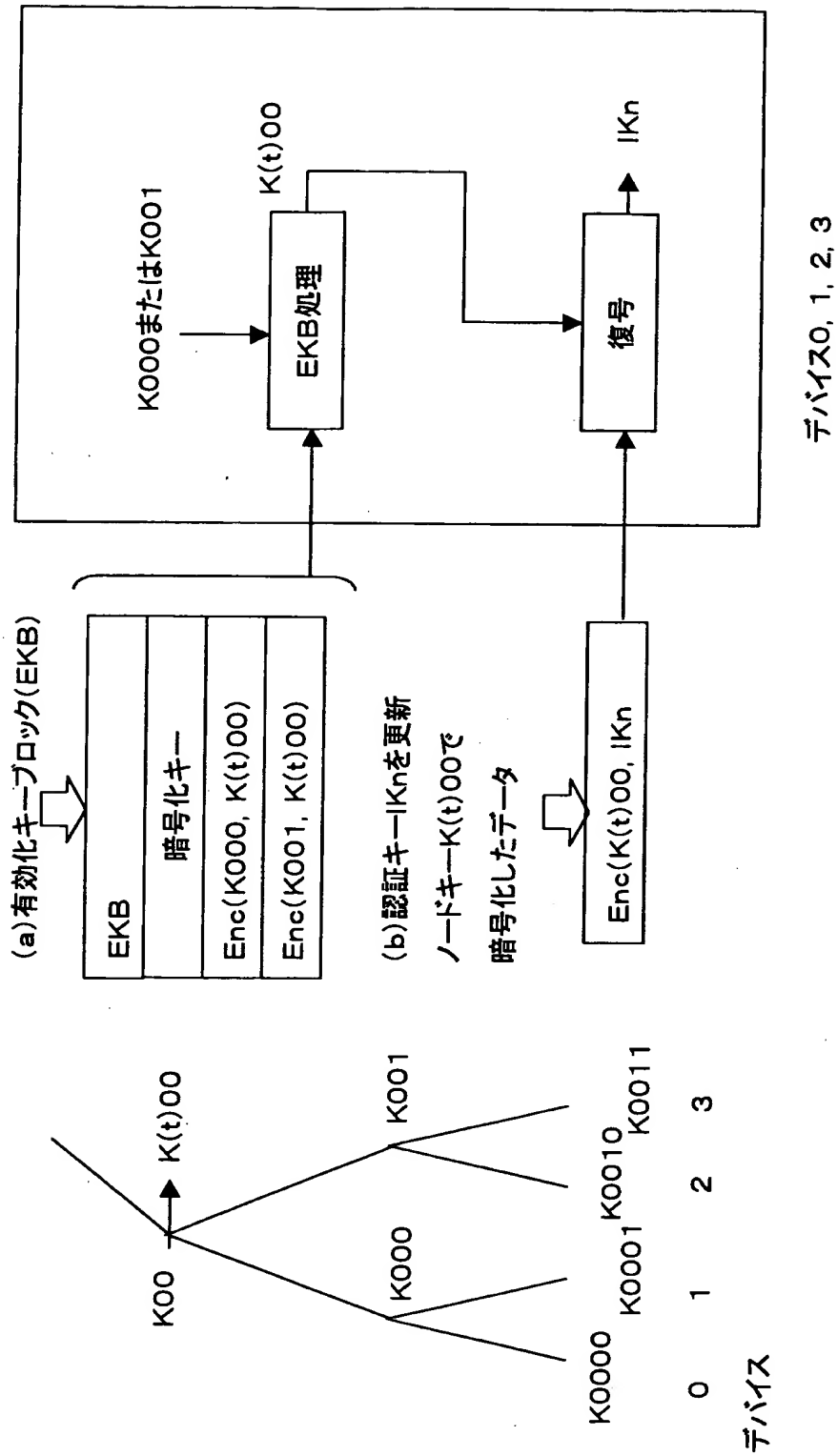
【図 3 2】



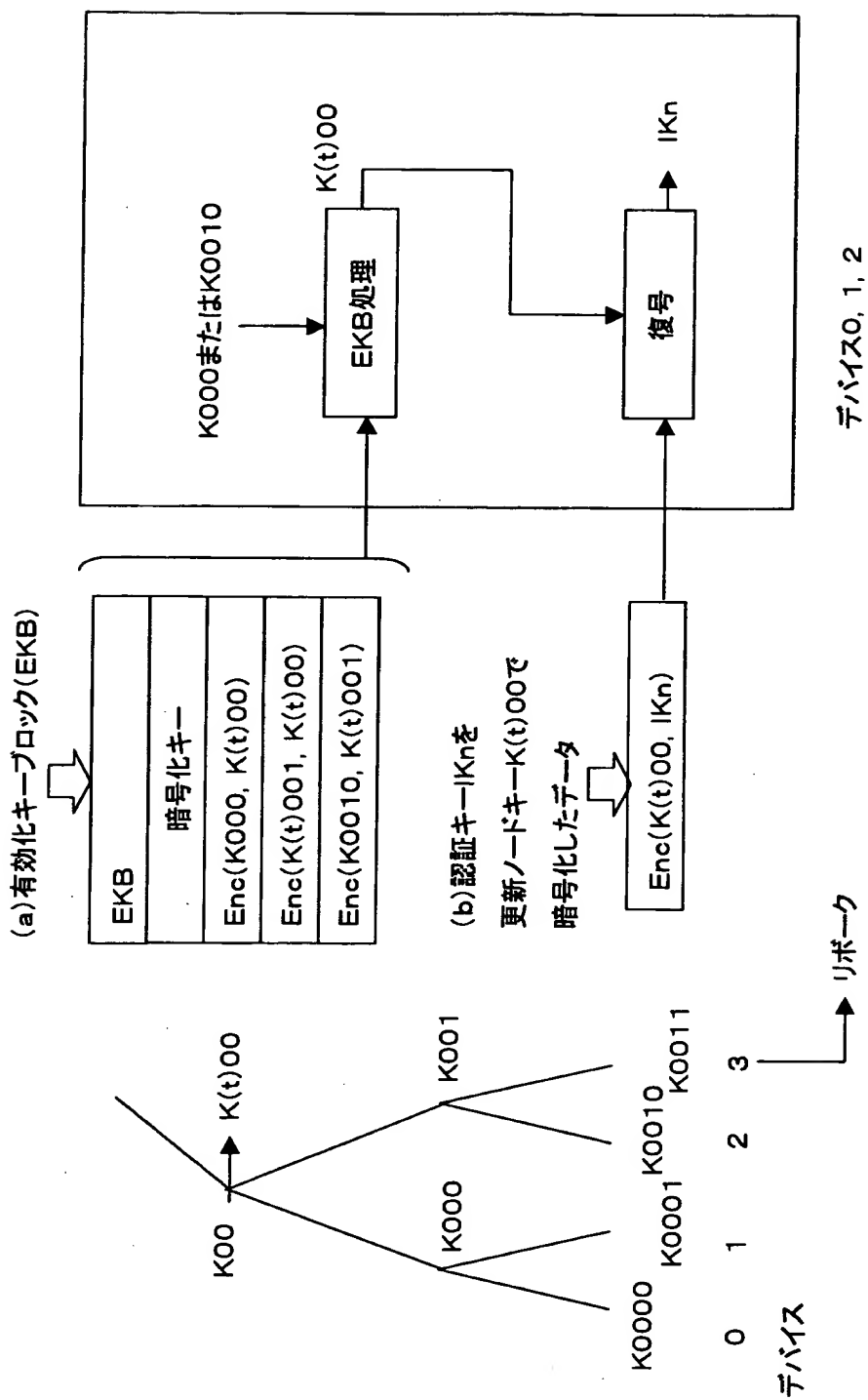
【図33】



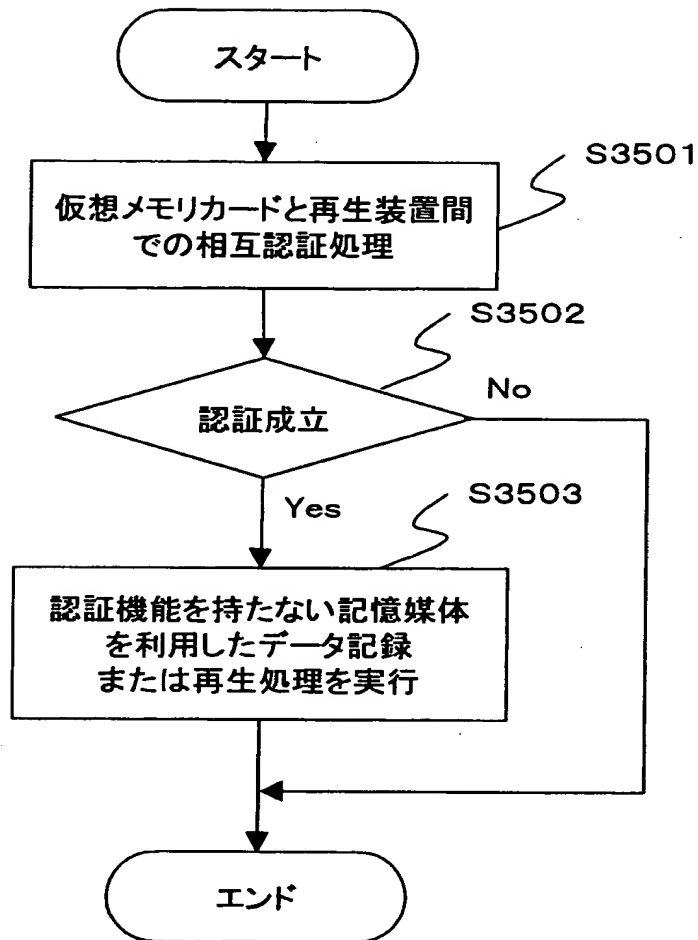
【図 34】



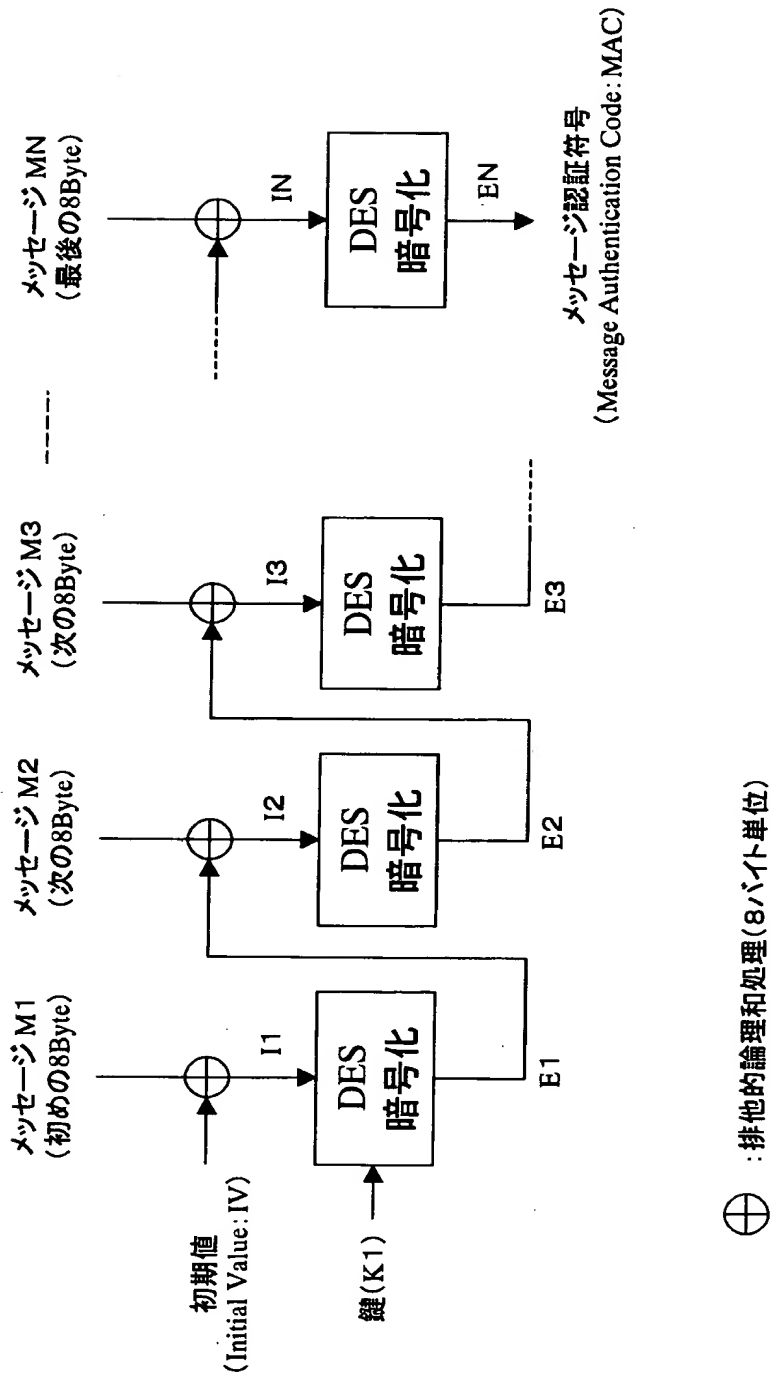
【図 35】



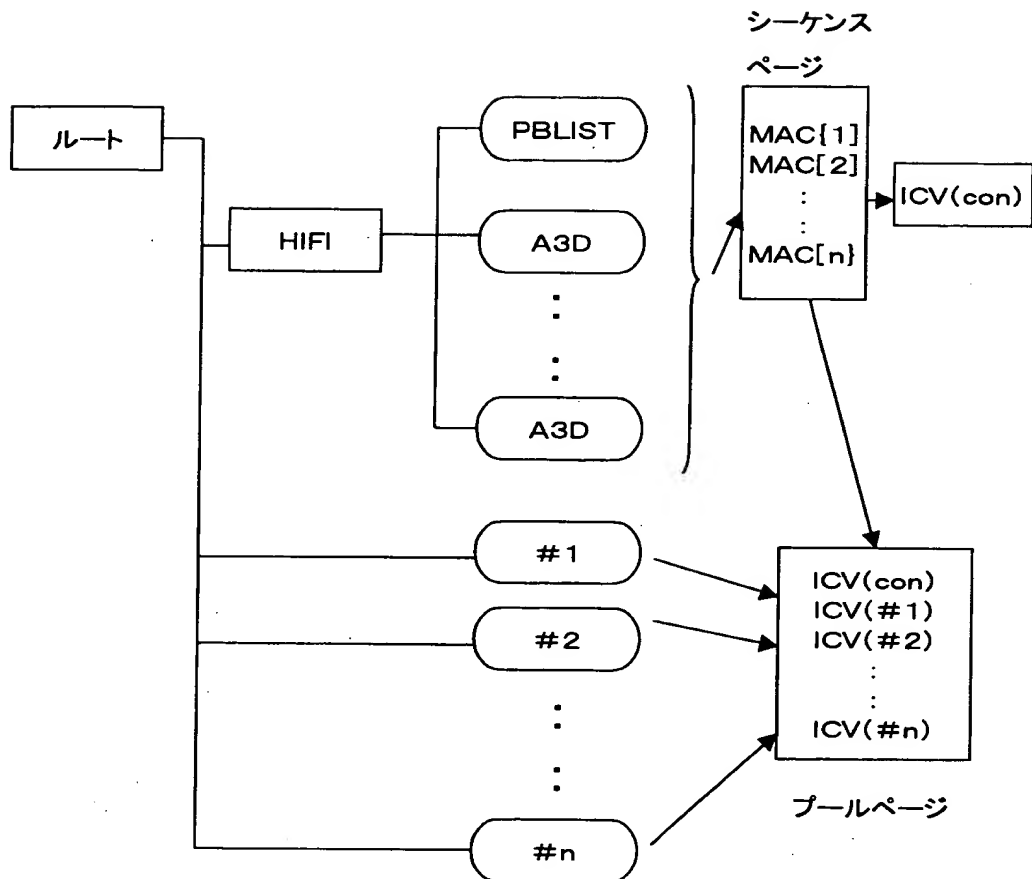
【図 3 6】



【図 3 7】



【図 3 8】



【図 3 9】

シーケンスページフォーマット

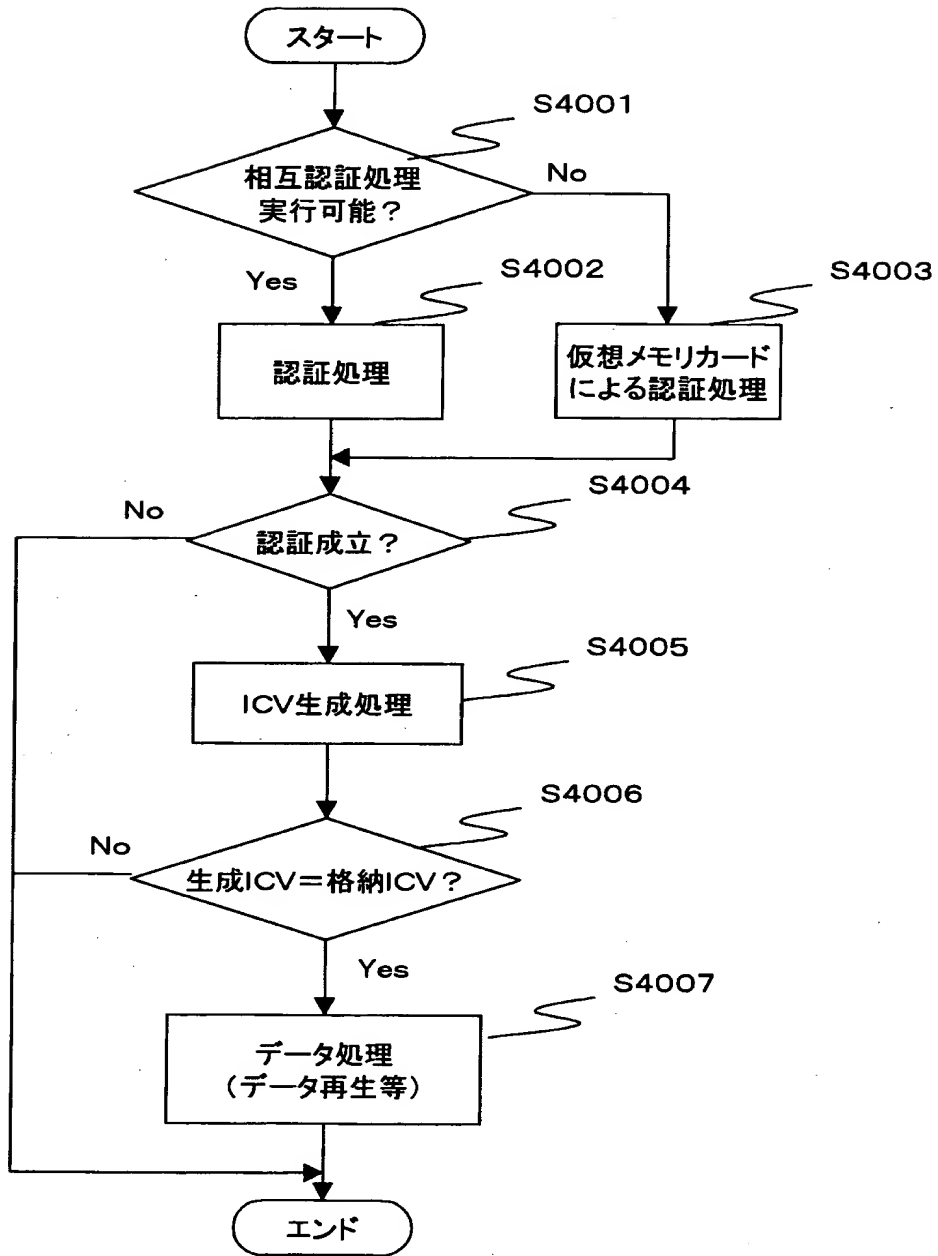
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x0000	E(Kstr, Kcon)								Reserved							
0x0010	ID(Upper)								IO(Lower)							
0x0020	C_MAC[0] (PUBLIST)								C_MAC[1]							
0x0030	C_MAC[2]								C_MAC[3]							
	:															
	:															
	:															
0x0FF0	C_MAC[nnn]								Reserved				Revision			

【図 4 0】

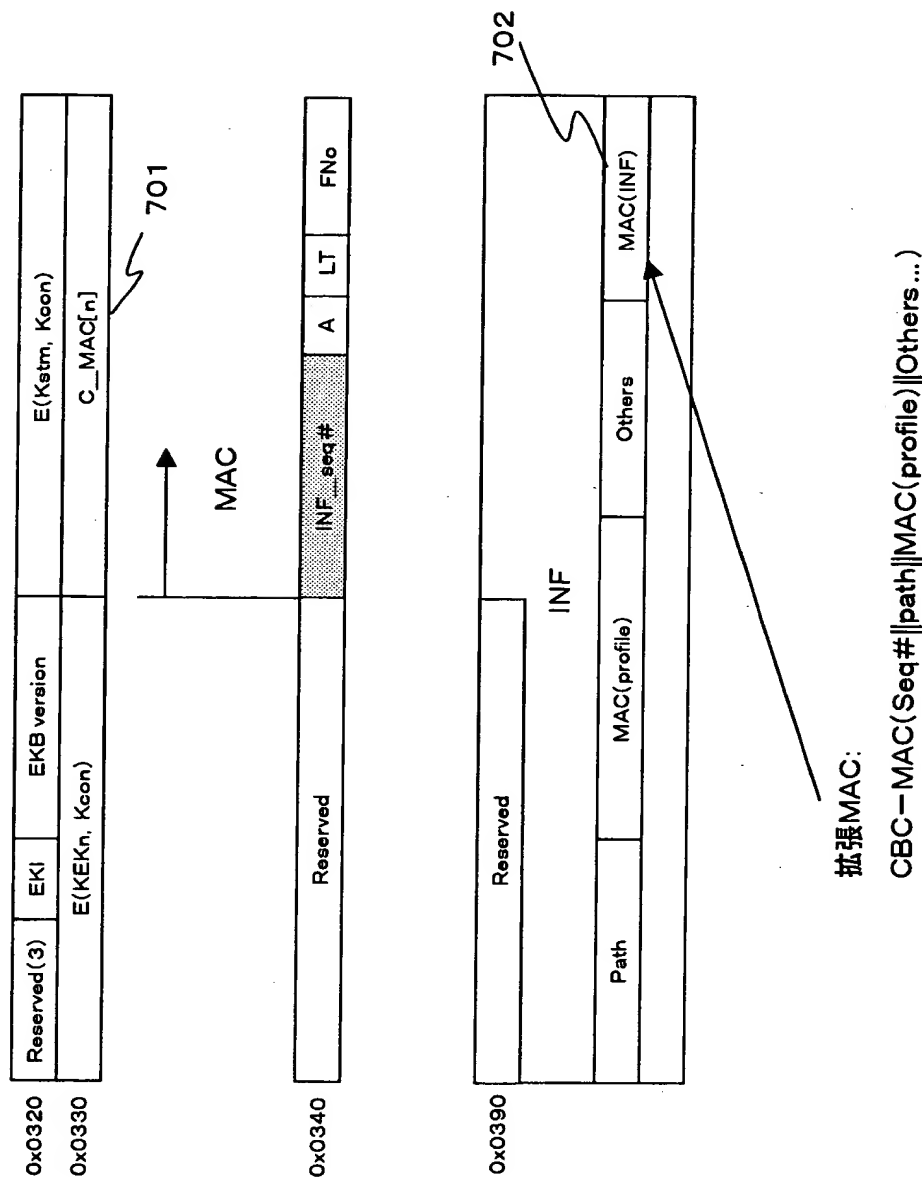
フルページフォーマット

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x0000	#0_revision			#0_EKB_version			#0_E(KEK, Kicv)									
0x0010	#0_E(KEK, Kicv)							ICV0								
0x0020	#1_revision			#1_EKB_version			#1_E(KEK, Kicv)									
0x0030	#1_E(KEK, Kicv)							ICV1								
	:															
	:															
	:															
	:															
	:															
	:															
0x01E0	#15_revision			#15_EKB_version			#15_E(KEK, Kicv)									
0x01F0	#15_E(KEK, Kicv)							ICV15								

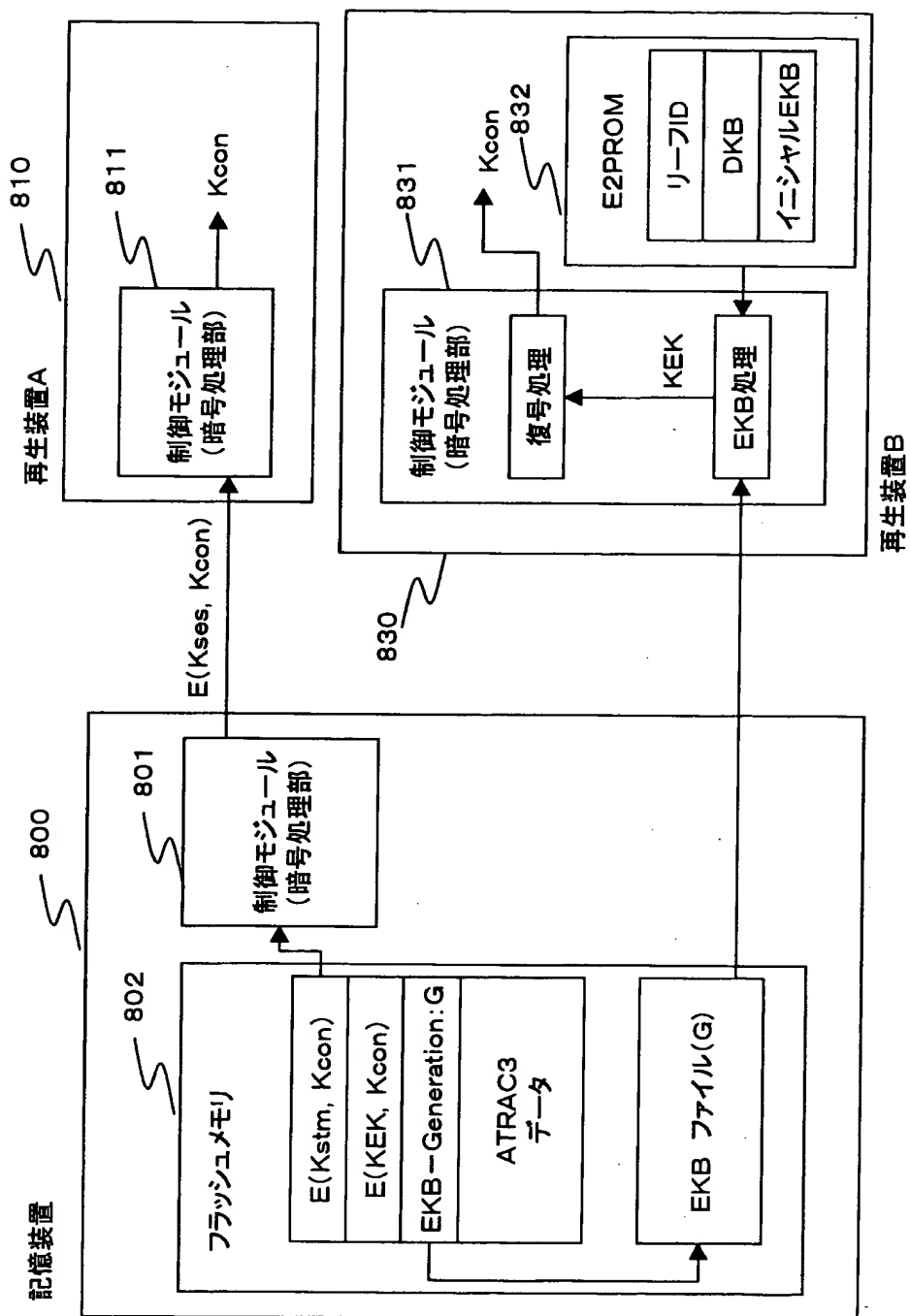
【図 4 1】



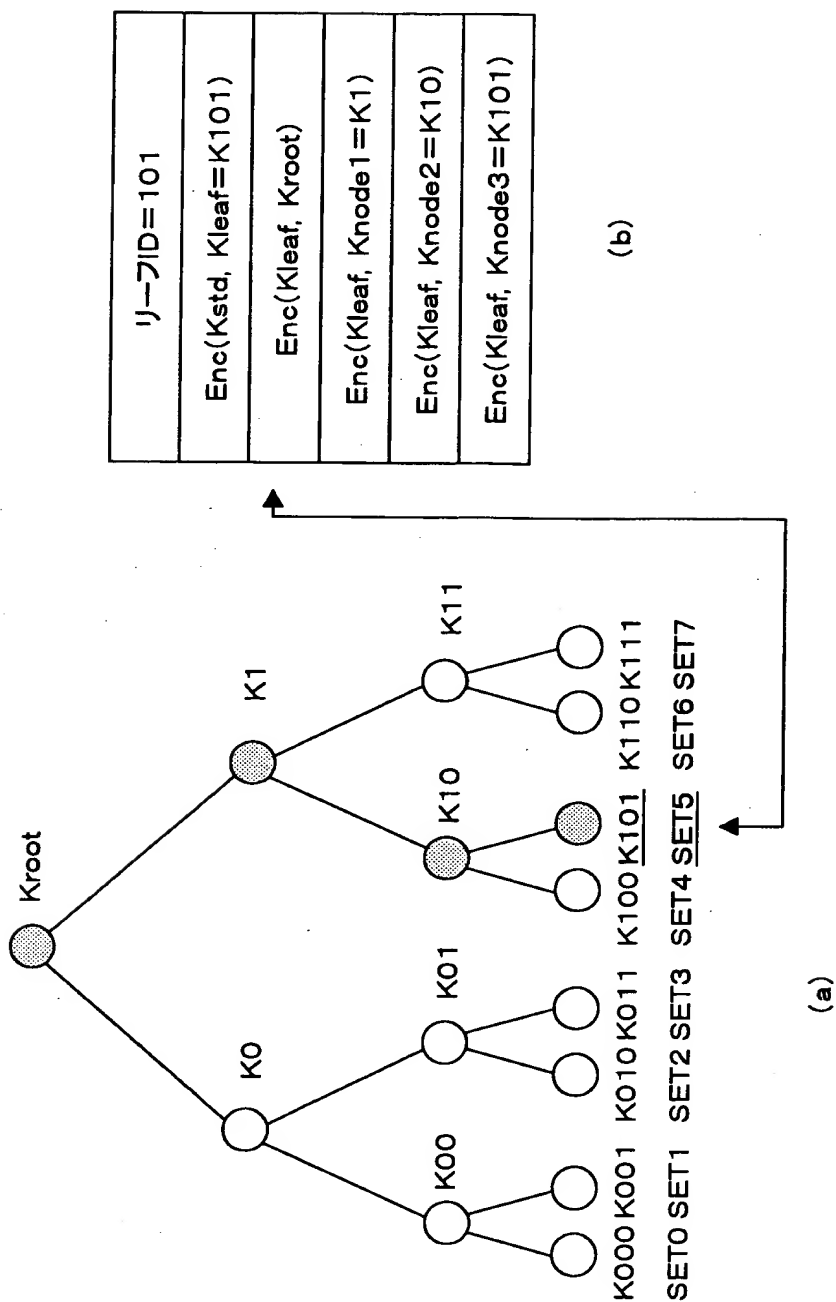
【図 4 2】



【図 43】



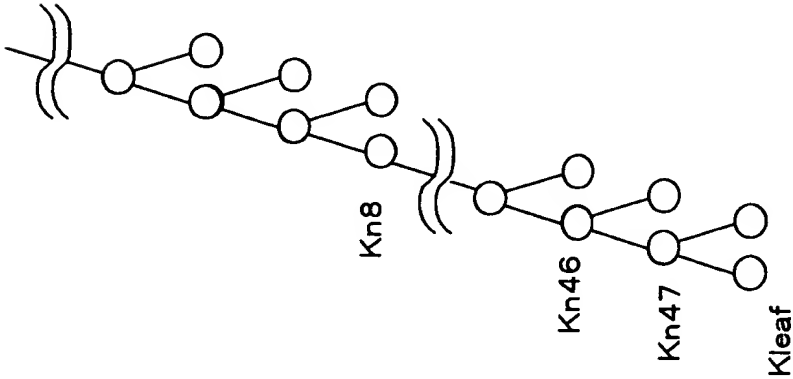
【図 4 4】



【図 4 5】

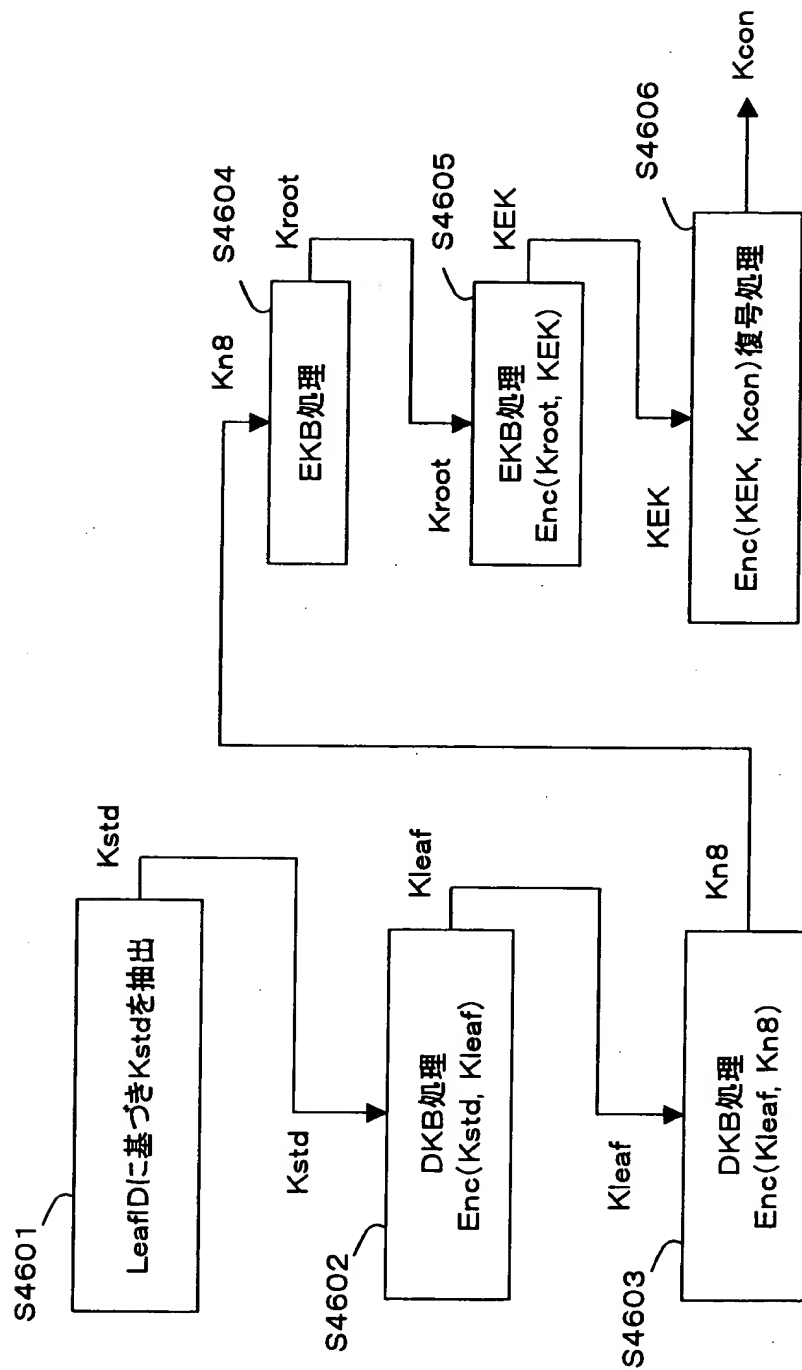
$I_j-ID=101$
$Enc(Kstd, Kleaf-1)$
$Enc(Kleaf, Kn47)$
$Enc(Kleaf, Kn46)$
$\vdots$
$Enc(Kleaf, Kn8)$
EKB

(b)



(a)

【図 4 6】



【書類名】 要約書

【要約】

【課題】 コンテンツの利用を行なうデータ処理装置におけるコンテンツ改竄チェックの効率化を実現した構成を提供する。

【解決手段】 記憶装置に格納されるコンテンツの検証値を生成してコンテンツに対応付けて格納し、コンテンツ改竄の有無を前記検証値により実行する構成において、検証値をコンテンツのカテゴリ毎に生成して格納する。カテゴリは、コンテンツの種類、あるいは、コンテンツの暗号処理鍵として設定されるコンテンツキー `K c o n` を暗号化して提供する有効化キーブロック (E K B) の管理エンティティ等に基づいて設定することにより、例えば有効化キーブロック (E K B) の管理エンティティ別に、コンテンツデータの検証処理を独立して実行可能となり、効率的な処理が可能となる。

【選択図】 図 3 8

認定・付加情報

特許出願の番号	特願 2000-247461
受付番号	50001044713
書類名	特許願
担当官	第七担当上席 0096
作成日	平成 12 年 8 月 22 日

<認定情報・付加情報>

【特許出願人】

【識別番号】	000002185
【住所又は居所】	東京都品川区北品川 6 丁目 7 番 35 号
【氏名又は名称】	ソニー株式会社

【代理人】

申請人

【識別番号】	100101801
【住所又は居所】	東京都中央区新富 1-1-7 銀座ティーケイビル 6 階 澤田・宮田・山田特許事務所

【氏名又は名称】	山田 英治
----------	-------

【選任した代理人】

【識別番号】	100093241
【住所又は居所】	東京都中央区新富 1-1-7 銀座ティーケイビル 6 階 澤田・宮田・山田特許事務所

【氏名又は名称】	宮田 正昭
----------	-------

【選任した代理人】

【識別番号】	100086531
【住所又は居所】	東京都中央区新富 1-1-7 銀座ティーケイビル 6 階 澤田・宮田・山田特許事務所

【氏名又は名称】	澤田 俊夫
----------	-------

出 願 人 履 歴 情 報

識別番号 [000002185]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都品川区北品川6丁目7番35号

氏 名 ソニー株式会社